TASKS AND VISUAL TECHNIQUES FOR THE EXPLORATION OF TEMPORAL GRAPH DATA

Natalie Kerracher

A thesis submitted in partial fulfilment of the requirements of Edinburgh Napier University, for the award of Doctor of Philosophy

March 2017

Abstract

This thesis considers the tasks involved in exploratory analysis of temporal graph data, and the visual techniques which are able to support these tasks.

There has been an enormous increase in the amount and availability of graph (network) data, and in particular, graph data that is changing over time. Understanding the mechanisms involved in temporal change in a graph is of interest to a wide range of disciplines. While the application domain may differ, many of the underlying questions regarding the properties of the graph and mechanism of change are the same.

The research area of temporal graph visualisation seeks to address the challenges involved in visually representing change in a graph over time. While most graph visualisation tools focus on static networks, recent research has been directed toward the development of temporal visualisation systems. By representing data using computer-generated graphical forms, Information Visualisation techniques harness human perceptual capabilities to recognise patterns, spot anomalies and outliers, and find relationships within the data. Interacting with these graphical representations allow individuals to explore large datasets and gain further insight into the relationships between different aspects of the data. Visual approaches are particularly relevant for Exploratory Data Analysis (EDA), where the person performing the analysis may be unfamiliar with the data set, and their goal is to make new discoveries and gain insight through its exploration. However, designing visual systems for EDA can be difficult, as the tasks which a person may wish to carry out during their analysis are not always known at outset. Identifying and understanding the tasks involved in such a process has given rise to a number of task taxonomies which seek to elucidate the tasks and structure them in a useful way.

While task taxonomies for static graph analysis exist, no suitable temporal graph taxonomy has yet been developed. The first part of this thesis focusses on the

development of such a taxonomy. Through the extension and instantiation of an existing formal task framework for general EDA, a task taxonomy and a task design space are developed specifically for exploration of temporal graph data. The resultant task framework is evaluated with respect to extant classifications and is shown to address a number of deficiencies in task coverage in existing works. Its usefulness in both the design and evaluation processes is also demonstrated.

Much research currently surrounds the development of systems and techniques for visual exploration of temporal graphs, but little is known about how the different types of techniques relate to one another and which tasks they are able to support. The second part of this thesis focusses on the possibilities in this area: a design space of the possible visual encodings for temporal graph data is developed, and extant techniques are classified into this space, revealing potential combinations of encodings which have not yet been employed. These may prove interesting opportunities for further research and the development of novel techniques.

The third part of this work addresses the need to understand the types of analysis the different visual techniques support, and indeed whether new techniques are required. The techniques which are able to support the different task dimensions are considered. This task-technique mapping reveals that visual exploration of temporal graph data requires techniques not only from temporal graph visualisation, but also from static graph visualisation and comparison, and temporal visualisation. A number of tasks which are unsupported or less-well supported, which could prove interesting opportunities for future research, are identified.

The taxonomies, design spaces, and mappings in this work bring order to the range of potential tasks of interest when exploring temporal graph data and the assortment of techniques developed to visualise this type of data, and are designed to be of use in both the design and evaluation of temporal graph visualisation systems.

Contents

List of Publications

Kerracher, N., Kennedy, J., & Chalmers, K. (2014). The Design Space of Temporal Graph Visualisation. In N. Elmqvist, M. Hlawitschka, & J. Kennedy (Eds.), *Proceedings of the Eurographics Conference on Visualization (EuroVis '14), Short Papers Track* (pp. 7–11). Swansea: The Eurographics Assocation. doi:10.2312/eurovisshort.20141149. **Awarded Best Short Paper**.

Kerracher, N., Kennedy, J., & Chalmers, K. (2015). A Task Taxonomy for Temporal Graph Visualisation. *Transactions on Visualization and Computer Graphics*, *PP*(99), 1.1. doi:10.1109/TVCG.2015.2424889.

Kerracher, N., Kennedy, J., & Chalmers, K. (2015). Visual Techniques to Support Exploratory Analysis of Temporal Graph Data. In E. Bertini, J. Kennedy, & E. Puppo (Eds.), *Proceedings of the Eurographics Conference on Visualization (EuroVis '15), Short Papers Track*. Cagliari: The Eurographics Assocation. doi: 10.2312/eurovisshort.20151133.

Bach, B., Kerracher, N., Hall, K. W., Carpendale, S., Kennedy, J., & Henry Riche, N. (2016). Telling Stories About Dynamic Networks With Graph Comics. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (pp. 3670-3682). ACM. doi: 10.1145/2858036.2858387.

Kerracher, N., & Kennedy, J. (2017). Constructing and Evaluating Visualisation Task Classifications: Process and Considerations. To appear in J. Heer, T. Ropinski and J. van Wijk (Eds.), *Proceedings of the Eurographics Conference on Visualization (EuroVis '17), Full Papers Track*. Barcelona: The Eurographics Assocation.

List of Figures

List of Tables

Acknowledgements

I would first and foremost like to express my deepest gratitude to my supervisors, Dr Kevin Chalmers and Professor Jessie Kennedy, who have been most generous with their advice, time, expertise, and patience when guiding me throughout the PhD process. This work could not have been completed without their support.

I would like to thank the School of Computing and Institute for Informatics and Digital Innovation for the support which made my studies possible, and Disney Research and the Scottish Parliament for their financial support and the opportunities to work with them.

I would also like to thank Natalia Andrienko for her insightful and helpful comments when constructing the task taxonomy in this thesis.

Finally, my thanks go to my family, friends, work colleagues, and fellow PhD candidates who have been supportive of me throughout my studies.

Dedication

I would like to dedicate this work to the people who have supported me throughout my studies, and without whom this thesis would not have seen the light of day. To my mum, dad, and sister who have always been there for me. To my Great Auntie Margaret whose kind words of wisdom have seen me through when times were tough. To my best friend Rich, who has been with me every step of the way. To David and our little bean. And finally, to my supervisors, Jessie and Kevin, whose guidance and support has made this work possible.

Chapter 1 Introduction

Temporal graphs are networks which change over time. Largely due to technological advances, there has recently been an enormous increase in the amount and availability of such data. Understanding the mechanisms involved in temporal change in a graph is of interest to a wide range of disciplines, from social and biological sciences, to computer networking, telecoms and transportation, to business and marketing. It is therefore important to have suitable tools to allow people to explore and analyse this kind of data. Information Visualisation (IV) – which has been defined as "*the use of computer-supported, interactive, visual representations of abstract data to amplify cognition*" ([1], p7) – provides useful techniques to support people in understanding what is happening in these networks.

1.1 Motivation

While static graph visualisation has a long history, recent years have seen a large increase in research directed toward the development of temporal graph visualisation systems: at the time of writing, Beck et al.'s digital library of publications relating to dynamic graph visualisation [2] contains 148 papers, the majority of which have been published since 2010. However, without referring to individual publications, it is difficult to understand how individual techniques relate to one another in terms of their similarities and differences, what types of analysis the different techniques support, and indeed whether new techniques are required for specific types of analysis.

When designing visual tools to support data analysis we need to consider both the characteristics of the data domain and the data, and the intentions and tasks of the people who will use the tools [3]. Visual approaches are particularly useful in supporting Exploratory Data Analysis (EDA), which focusses on detecting and describing patterns and relationships in the data. When carrying out EDA, the person performing the analysis may be unfamiliar with the data, and – at outset at least –

they may have no specific goal in mind other than to further their understanding of, make new discoveries about, and gain insight into the data set [4]. The analysis process typically proceeds in an iterative fashion: gaining some general overview of the data, spotting something interesting, investigating in further detail, formulating new questions based on what was discovered, and so on. However, as the tasks are not necessarily known at the outset, system designers must anticipate the potential tasks in order to make an informed decision regarding which tools to include, and to ensure that a sufficiently wide range of tasks are supported [5]. An understanding of the potential range of tasks that may be involved is therefore required when designing tools and systems.

In addition to the role played in the design of visual tools, task understanding is also required when evaluating tools and systems in terms of user performance and user experience [6]. Identifying and understanding tasks has resulted in a relatively new branch of information visualisation research and given rise to the creation of a number of task taxonomies in order to structure these tasks in a useful way [7].

No single visualisation technique is able to support all types of data and all of the tasks that an individual may wish to carry out during data analysis. An understanding of which techniques are able to support which data types and tasks is therefore also required when designing and evaluating visual tools and systems. In the early days of visualisation research, Wehrend and Lewis [8] proposed a "cataloguing" method which classifies general visual techniques by the tasks which they support. The resulting catalogue was intended to support sharing of methods across different domains, by helping system designers to identify the visual techniques capable of supporting the tasks for which they were designing. Twenty five years have passed since this approach was first proposed, and a vast number of Information Visualisation tools and techniques have been developed in that time. While organising the tools and tasks for all types of data in this way would now be an enormous undertaking, in this thesis a similar approach is proposed for only a subset of data types, specifically temporal graph data.

Given the importance and increasing availability of temporal graph data across a wide range of domains, and the need for - and current interest in - developing visual techniques to support the analysis of such data, this thesis seeks to address the strong need to understand both the tasks associated with exploratory analysis of temporal graph data, and the visual techniques which are able to support these tasks. In so doing, a number of tools are developed to bring order to the range of tasks and existing visual techniques, identify opportunities for research, and assist in the design and evaluation of temporal graph systems.

1.2 Research Questions

This work addresses four key questions:

1. What are the possible exploratory analysis tasks that temporal graph visualisation might need to support?
2. Which visual techniques, tools, and approaches, have been developed to support exploration of temporal graph data? Are there any unexplored opportunities for visual techniques?
3. Which visual techniques support which types of tasks?
4. For the tasks identified in (1), are there suitable visual techniques or are new/better visual techniques required?

To address question 1, an existing generic task framework is extended for use with graph data, and a design space of temporal graph tasks is developed. This is the subject of Chapter 4 and Chapter 5, with an evaluation of the work presented in Chapter 6.

For question 2, the literature relating to temporal graph visualisation is surveyed, from which two dimensions for categorising tools and techniques are extracted (graph and time encodings). These dimensions are used to construct a design space of possible temporal graph visualisation techniques. Existing systems, tools, and techniques are mapped to this design space, revealing possible techniques which are less well explored. This work is presented in Chapter 7.

Having identified the task-space and technique-space, in order to address question 3, the mapping between them is considered in Chapter 8. Given the number of individual tasks identified, the task dimensions are used as a framework for the discussion of task support. A somewhat surprising - yet important - finding made during the task-technique mapping stage is that tools and techniques from a much wider range of research areas than just temporal graph visualisation are required to support the identified tasks. Techniques from temporal visualisation, static graph visualisation, and visual comparison, are therefore also reviewed and included in the task-technique mapping.

Through the process of considering visual techniques for the support of the identified temporal graph tasks, a number of gaps in task coverage are identified, along with tasks which are less well considered in the literature, answering question 4. Developing techniques to support these tasks could prove interesting avenues for future research.

The use of the developed tools in the design and evaluation processes is demonstrated through an empirical study and case study. The empirical study in Chapter 6 evaluates the task framework's use in overcoming a key problem of designing for EDA outlined earlier: that visualisation designers need to anticipate the tasks which the people they are designing for may want to perform, but these people may be unable to articulate all of the tasks which are of interest to them at outset. It explores the use of the task framework at the task understanding stage of the design process, in helping to discover the analytical tasks which may be of interest. In the case study of Chapter 9, using a methodology derived from the task framework an existing temporal graph visualisation tool is evaluated in terms of its existing task support. Using the task-technique mapping, recommendations are made for the inclusion in the system of alternative and additional techniques in order to improve task coverage.

1.3 **Contributions**

The contributions of this work are fourfold:

(1) A characterisation of temporal graph data and tasks, including:

- An extension to an existing formal task framework, to handle graph data.
- A temporal graph task taxonomy, produced by applying the extended framework to temporal graph data.
- A design space of potential temporal graph tasks, produced by combining the identified task dimensions using a series of matrix structures.

By illuminating the potential tasks involved in exploratory analysis of temporal graph data, such a characterisation will be of benefit to designers of visualisation systems in assisting with task specification during requirements analysis. It will also be of benefit to evaluators, by representing the range of tasks for inclusion in evaluations, and providing grounds on which to justify the inclusion of specific tasks when designing evaluations.

(2) A characterisation of temporal graph visualisation techniques, including:

- A review of existing systems, tools, and techniques for temporal graph data
- A design space of possibilities for temporal graph visualisation techniques
- A mapping of existing techniques to the design space

The mapping of existing literature to the design space brings order to the range of existing systems, tools, and techniques spread across different domains. It also reveals less explored and unexplored possibilities, which could prove interesting avenues for the development of novel visualisation techniques for representing temporal graph data.

(3) A review of techniques to support temporal graph tasks, revealing less well supported and unsupported tasks.

Such a classification will be of use in both the design and evaluation of temporal graph visualisation systems. In terms of design, it will offer guidance to designers in choosing the most appropriate tools for inclusion in the design of a system, based on the tasks the system is required to support. While the tasks are purposefully domain independent, the techniques are drawn from across a wide range of disciplines, thus, techniques may come from areas with which a designer would be otherwise unfamiliar. In addition, because the taxonomy is extended from a generic framework, the tasks are much wider than those which most temporal graph visualisation techniques currently consider: the need for inclusion of techniques from general, temporal and static graph visualisation research areas when developing systems is therefore highlighted. The mapping may also draw attention to opportunities for evaluation: where more than one technique supports a single task, it can then be investigated which technique performs best. Finally, a number of less explored and unexplored tasks are revealed, which could prove worthwhile avenues for future research.

(4) A review of classification construction and evaluation practices

Categorising tasks is a common pursuit in visualisation research, with a variety of taxonomies, typologies, design spaces, frameworks, and models having been developed over the last three decades. The usefulness of these classifications in the design and evaluation processes is also widely accepted. However, while evaluation practices have also become a topic of increasing interest in the visualisation community e.g. [6], [9], very little attention has so far been given to the construction and evaluation practices involved in developing task classifications. While we would expect a publication demonstrating a new visualisation technique or system to include some form of evaluation with respect to its utility, performance, and limitations, this does not appear to be the case when newly developed classifications are reported. Further, while much work reflects on and provides guidance relating to appropriate design and evaluation practices when developing visualisation systems and techniques (e.g. [6], [9]–[12]) analogous guidance for developing task classifications does not exist. This is surprising, given that measuring the effectiveness of classifications has been recognised as a difficult problem [13], and the benefits of evaluating classifications are comparable to those of evaluating systems, including identifying areas for improvement resulting in better classifications, convincing potential adopters of the validity and utility of the approach (particularly important for more complex classifications which may require significant effort to adopt), and helping adopters select between competing classifications.

A final contribution of this thesis therefore is its elucidation of the classification construction process, the threats to validity at each stage of construction and means of mitigating these threats, along with detailed consideration of the appropriateness of evaluation strategies according to the different aspects of the classification which they seek to evaluate. While the work stops short of providing prescriptive advice on constructing and evaluating classifications, the guidance arising from these investigations will be of benefit to developers of classifications in determining appropriate construction and evaluation strategies when developing a classification, and also for those selecting between competing classifications for use in the design and evaluation processes. This contribution can be found in Sections 3.1 and 6.1.

1.4 Thesis Structure

The thesis is structured as follows:

Chapter 2 provides an overview of the temporal graph visualisation research area. It considers the role that taxonomies play in visualisation research and development generally, and reviews work to date in developing task and visualisation technique taxonomies for temporal graph data.

In **Chapter 3** a review of the task classification literature is presented which considers the stages of classification construction and identifies the associated threats to validity arising at each stage and in response to the different construction methods employed. Guidance is offered on suitable validation approaches in order to mitigate these threats. An overview of the Andrienko task framework [5] - on which the task taxonomy in this work is based - is presented, along with a discussion of the limitations of their framework when used with graph data, which necessitated a number of extensions.

Chapter 4 presents the extensions to the Andrienko task framework necessary to overcome its limitations in the graph case.

Based on the extended version of the task framework outlined in Chapter 4, **Chapter 5** presents a task taxonomy for temporal graph. The dimensions of the taxonomy are combined using a series of matrix structures to produce a design space of potential tasks associated with the exploration of temporal graph data.

Chapter 6 focusses on the evaluation of the developed task framework. In order to determine a suitable evaluation strategy, existing evaluation approaches in the visualisation literature are reviewed. The various aspects of classifications which can be evaluated are distinguished and appropriate evaluation methods are considered in relation to these aspects. Based on the findings of this work, the task framework is evaluated firstly in relation to other extant classifications, and secondly via an empirical study which demonstrates its use in the design process.

Chapter 7 reviews existing visualisation techniques for representing temporal graph data. A two-dimensional design space (based on time and graph structural encodings) is developed to categorise temporal graph visualisation techniques. Existing tools and techniques for temporal graph visualisation are mapped to this design space, revealing a number of unexplored possibilities for visual representations.

Chapter 8 explores the visual techniques which are appropriate for the support of each of the task categories of the framework. Through this task-technique mapping, a number of less well supported and unsupported tasks are identified.

Chapter 9 presents a case study in which the tools developed in this thesis were used to evaluate an existing temporal graph visualisation system and make design recommendations to improve the system's task coverage.

Chapter 10 summarises the main contributions of the work and considers future directions.

Chapter 2 Background

This chapter considers the role that taxonomies play in the design and evaluation of visualisation systems and techniques. Existing work in developing task taxonomies and visualisation technique taxonomies is reviewed, both generally, and specifically relating to temporal graphs. A discussion of the areas requiring further work - and how the work in this thesis seeks to address those areas – is also included. The chapter begins with a characterisation of temporal graph data and an overview of key challenges within the research area of temporal graph visualisation.

2.1 Temporal graph visualisation

Formally, a graph, G = {V, E}, consists of a set of vertices or nodes, V, and a set of edges, E, which connect pairs of vertices. From organisational structures to biological networks and transportation routes, graphs have been used to model relational data from a wide range of domains. Static graph visualisation techniques have long been used to represent the relational aspects of data, helping people understand the ways in which entities in the data are connected, and the larger structural patterns that individual connections produce.

There are two main ways in which graphs are visually represented: node-link diagrams and matrices. Node-link diagrams have a long history and are the most commonly employed representation: nodes are represented by some sort of shape, which are connected together with lines representing edges. The key challenge for this type of representation lies in laying out the graph, in other words calculating where to position nodes, in order to produce a 'good' graph layout i.e. one which is readable and clearly understood. A set of aesthetic criteria - or rules for good graph layout - have been defined, which include minimising edge crossings and edge length, distributing nodes evenly, and not allowing nodes to overlap [14]. Graph layout algorithms are used to calculate node positions and take into account these aesthetic criteria. However, some of the aesthetic criteria are mutually incompatible, requiring trade-offs to be made, while satisfying even an individual aesthetic criteria can be an NP-hard problem. Computing node positions while satisfying the aesthetic criteria can therefore be computationally expensive. Moreover, computation time and complexity increases as the graph becomes larger. Much of the work in static graph visualisation has therefore been directed at developing layout algorithms.

Temporal graphs (also known as dynamic graphs), are graphs which change over time. Temporal graph data is ubiquitous: changing relationships in social networks, traffic flows in transportation networks, gene regularity networks in biology, connections between machines in computer networking, calls between subroutines in software systems - these are just a few examples of graph structures which change over time. There are two broad categories into which the changes can be grouped: structural change (e.g. nodes or edges being added and/or deleted) and attribute change (e.g. an increase, decrease, or categorical change in node attribute values or edge weightings). The primary concern of temporal graph visualisation is communicating these changes.

Temporal graph visualisation is a relatively new research area in the well-established field of graph drawing, and there is still much ground to be explored. While most graph visualisation tools focus on static networks, there has been a recent increase in the number of tools focusing on temporal graph visualisation: Beck et al.'s dynamic graph visualisation literature database [2] contains references for 52 such applications, the vast majority of which appeared post-2003. Little attempt has been made to extract the unique visualisation techniques used in temporal graph visualisation applications, or to establish which tasks the different techniques are able to support.

The main focus in the temporal graph literature to date has been on visualising structural change, as existing static graph visualisation techniques can be combined with, for example, colour encoding on nodes or edges to indicate change in attribute values [15]. In addition to the challenges for static graph drawing there are a number of further challenges for temporal graph visualisation.

Firstly, time adds an additional layer of complexity to the problem of computing node positions. In addition to computing a layout which satisfies the aesthetic criteria at a single point in time, we must also maintain the internal understanding of the graph, or 'mental map', of the person using the visualisation, as the graph changes over time. This generally requires minimising unnecessary change in the visualisation to avoid disrupting the mental map, whilst showing the changes occurring in the data. Work in this area can be divided into two main areas: usability studies focussing on peoples' understanding of changes in the graph (such as [16]–[21]) and the computational difficulties of adapting and developing layout algorithms for dynamic graphs (e.g. [22]–[26]). When developing such layout algorithms, a distinction is drawn in terms of 'online' and 'offline' data, as this affects the computational difficulty of the layout algorithm and restricts layout possibilities. Offline data is where all of the data is known beforehand. This makes it possible to compute a 'supergraph' of all 'timeslices' (snapshots encoding the structure of the graph at a given time [27]) which can be used as a base for computing stable layouts. Online data is not known beforehand and may be continually added to. This means that layout algorithms can only take into account previous timeslices when computing layouts [28].

An additional challenge for temporal graph drawing is how to visually encode the temporal dimension, given that the two spatial dimensions are normally used to lay out the graph to show its structure [29]. A number of possible approaches to encoding the temporal aspect of the data have been proposed, each of which have relative strengths and weaknesses. The different approaches to encoding the temporal dimension are explored in detail in this thesis.

2.2 **Taxonomy**

The word "taxonomy" can be used to describe both the general science of classification, and a particular system of classification. Taxonomy is used to classify large corpora of information. It is a way to organise, synthesise, and contextualise knowledge. It goes beyond simple classification of like items into groups, by describing the relationships that exist between items. The aim of taxonomy is to ensure that items within a group are as similar as possible, while ensuring that the groups themselves are as distinct as possible; in other words, minimising within-group variance while maximising between-group variance [30]. Bringing order to a disorganised body of knowledge in this way can help us make sense of complex subject matter and large collections of items.

Taxonomy is employed across a wide range of domains, but has its roots in biological sciences, where its modern meaning dates back to Carolus Linnaeus (1707-78), and specifically refers to the classification of biological organisms into hierarchical sets on account of their shared characteristics [31]. Understanding which organisms are similar and which are distinct is key to many biological pursuits, from managing pests to ensuring safety in herbal medicine; Smith et al. [32] collected together 48 case studies demonstrating the use and importance of taxonomy in biology.

Taxonomy is also widely used in information systems and knowledge management. In content management systems, taxonomies are used to provide structured navigational paths through content collections. Taxonomies for search engines help improve the relevance of search results. The Dewey Decimal Classification system – said to be the world's most widely used taxonomy – is a general knowledge organisation tool used to organise library materials by discipline.

Other areas in which taxonomy is employed include chemical classification, psychology (e.g. Moffitt's taxonomy of anti-social behaviour [33]), engineering (e.g. Gershenson's taxonomy of corporate requirements which impact design requirements [34]), and education (e.g. Bloom's taxonomy of educational objectives [35]).

2.3 The role of classification in visualisation research

Categorising tasks and visual techniques is a common pursuit in the visualisation research community. Various taxonomies, typologies, design spaces, and frameworks, have been developed over the last three decades. These have been used to pre-empt and make sense of both the aims and intentions of the people using visual analysis techniques, and the ever increasing literature relating to visual tools, techniques, and systems to achieve these aims.

Classifications provide a useful means for bringing order to the range of existing visualisation systems, tools, and techniques, often from across a wide range of domains. They can also act as an entry point for researchers new to an area, in a manner similar to review and survey papers. Moreover, they play a useful role in the design and evaluation of visualisation tools and systems. Before looking more closely at the uses of classifications, let us first consider the terminology used in the visualisation literature to describe these constructs.

2.3.1 Terminology

Classification can be used as an umbrella term to describe a construct in which items are collected together and grouped in some meaningful way. However, many terms are used in the visualisation literature to describe such constructs. *Lists* of tasks are usually intended to be non-exhaustive illustrations of exemplar or common tasks, which may or may not be grouped into categories (e.g. [36]). The terms *typology* and *taxonomy* tend to indicate a more rigorous process of categorisation has been followed, and are often used interchangeably in the literature. However, Bailey [30] distinguishes them on the grounds that a taxonomy is empirical (a set of existing entities are grouped according to their similarity to produce a classification), while a typology is conceptual (a classification is constructed a priori using multiple conceptual dimensions; the resulting categories represent concepts rather than empirical cases). Such classifications are used in the visualisation literature to describe and bring order to the range of tasks and visual techniques. Bailey [30] also describes the case where the independent dimensions of a classification are combined to form a *property space*. In the visualisation literature, this idea – often termed *design space* - is becoming increasingly common [37]. The intended use of a design space is not simply as a means to classify existing items, but to map "*the space of the possible*"[37], revealing potential items which may not yet exist. As such, it can be used as a generative method to specify novel visualisation techniques (e.g. [38], [39]) or previously unconsidered tasks (e.g. [37]).

Task classifications are often used to *characterise systems* according to the tasks they support in order to help make comparisons between systems when selecting appropriate visual tools (e.g. [40], [41]). An extension of this idea are *task-technique mappings* or *catalogues* (e.g. [42]) which take a task classification and map to each category the visual techniques for their support. These are often intended to provide a useful inventory of appropriate techniques for use during the design process, and like design spaces, can help point to opportunities for research by identifying as yet unsupported tasks which could benefit from the development of appropriate visual techniques.

2.3.1 Role in communication

Perhaps the most fundamental benefit of classifications is that they provide a common vocabulary to describe both analytical tasks and the visual means by which they can be achieved (objectives and actions, respectively, to use the language of Rind et al. [43]. Having an agreed upon language allows researchers to communicate more effectively, reducing misunderstanding [37], [44]. As discussed further in Sections 2.3.4 and 2.3.5, using classifications which present tasks in a consistent and abstract manner to describe the domain specific tasks of users, and the functionality of systems and tools from across domains and application scenarios, offers many benefits in the design and evaluation processes. Describing tasks in an abstract rather than domain specific manner also allows us to generalise when situating and communicating the results of research. For example, Lee et al [40] suggest the use of their classification in helping evaluators generalise results collected in a series of controlled experiments. Sedlmair et al. [12] note the need to present clear abstractions of tasks when reporting on design studies, so that the bare minimum of domain knowledge is required to understand them. Similarly, Rind et al. [43] note the use of abstract tasks when setting context in case studies. Using the recognised terminology of task classifications can be particularly beneficial in these circumstances.

2.3.2 Making sense of what's out there

Classifications can help us make sense of what already exists in our research area. They provide a useful means for bringing order to the range of user intentions and existing visualisation systems, tools, and techniques, often from across a wide range of domains. They can act as an entry point for researchers new to an area, in a manner similar to review and survey papers. Virtually all classifications are developed with this purpose in mind.

2.3.3 *Identifying what's not out there*

Schulz et al. [45] describe how design spaces can be used to identify "*the space of the possible*". A design space maps out the "*universe of all possible design choices*" [38] and can be constructed by combining the independent dimensions of a taxonomy to produce all possible variants. Design spaces have been used to map the space of the possible for both visual techniques e.g. [38],[39], and tasks e.g. [41], [45]. By mapping existing techniques to the possibilities identified in a technique design space, as-yet unexplored techniques may emerge, which could prove interesting opportunities for further research e.g. [38]. Further, visual techniques can be mapped to a task design space according to the categories of tasks which they support, to establish which tasks are currently addressed by existing techniques, and reveal areas which could benefit from further research e.g. [41]. In this way, mapping techniques to the 'space of the possible' can help guide the focus of future system development and encourage the pursuit of novel research questions [41]. Such a mapping could also help signpost opportunities for evaluation, as it identifies the case where multiple techniques support the same task: these techniques are potential candidates for use in controlled experiments to establish which techniques are the most effective in their support.

2.3.4 *Use in the Design Process*

Several authors note the role which task classifications can play in systematising the design process. Both Amar and Stasko [46] and Sedig and Parsons [47] note that classifications can act as a systematic basis for thinking about the design process, while the use of classifications as a "checklist" of items to consider when designing visualisation tools is proposed by [46], [48], [49].

Classifications of tasks and techniques can be gainfully employed at multiple stages of the design process. Before discussing the use of classifications in the design process, let us briefly consider the design process itself.

While the field of visualisation draws heavily on practices in other domains for guidance on designing visualisation systems (HCI, engineering, design, etc.), specific models have recently been developed within the community to help describe and facilitate this process. Possibly the best known is Munzner's nested model [10], which outlines the four major stages at which design decisions need to be taken in the process of designing a visualisation, and the threats to validity and validation required at each level. The four stages identified are:

1. *Domain problem characterisation*: this is the requirements gathering stage, where the visualisation designer learns about the domain specific tasks and data of the target users.

2. *Data/operation abstraction design*: at this stage, the domain specific tasks and data are mapped to the vocabulary of information visualisation.

3. *Encoding/interaction technique design*: appropriate visual encodings and interaction techniques are selected at this stage.

4. *Algorithm design*: this stage deals with the design of the algorithm which automatically encodes the data in the selected manner.

Meyer et al. [11] extend this model by introducing 'blocks' (outcomes of the design process at each level) and 'guidelines' (which describe the relationships between blocks). McKenna et al. [50] further build on these models (and Sedlmair et al.'s nine stage design study methodology [12]) by introducing four 'design activities' – understand, ideate, make, and deploy – which map to the four levels of the nested model Figure 1. For each design activity they describe the associated motivations, outcomes, and methods. Their paper lists over 100 exemplar methods which may be utilised at the different stages, including well known techniques such as interviewing and controlled experiments, and perhaps less commonly used methods such as graffiti walls and love/breakup letters. In addition to classification by activity, methods are also classified according to whether they are generative (those which are intended to be divergent and create many outcomes e.g. brainstorming) and/or evaluative (those which are convergent and used to filter outcomes e.g. feedback from user studies).

	Understand	Ideate	Make	Deploy
Domain characterisation	x			
Data/task abstraction	x	x		
Encoding/interaction technique		x	x	
Algorithm design			x	x

Figure 1 Mapping of McKenna et al.'s [50] design activities to four levels of Munzner's [10] nested model. Redrawn from [50], Figure 2.

The following discussion considers the use of classifications in the design process with respect to the first three stages of Munzner's nested model [10].

2.3.4.1 Task understanding

Understanding which analytical tasks an analyst may wish to carry out is a non-trivial problem and a key component at the domain problem characterisation stage. Despite the recognised importance of this stage, and calls for further work in this area e.g. [10], [51], this stage is known to be under-researched in the visualisation community [13], [9], [11].

In a typical design scenario, van Wijk [52] notes that visualisation researchers must spend time and effort bridging 'the knowledge gap' between themselves and the domain expert, in order to effectively understand the problem in what is potentially an unfamiliar domain with its own terminology. Generative methods [50] for eliciting possible tasks of interest can be roughly grouped into three strategies: deriving tasks in an analytical fashion, for example, by reviewing relevant literature; talking to domain experts, for example, through interviews or brainstorming sessions in focus groups; observing people at work, either using existing visualisation tools or the methods they currently employ.

As discussed further in Section 3.1.1.1, each of these strategies has limitations when eliciting tasks. One use for task classifications is in supporting the generative phase of task understanding in order to mitigate some of the problems in the strategies outlined above. For example, they may act as a useful means upon which to base discussions with domain experts. By setting out the range of potential tasks of interest, they may overcome known problems associated with simply asking people to introspect. They may also help to keep the discussions focussed on tasks; one pitfall identified by Sedlmair et al. [12] at this stage of the design process is allowing experts to focus on possible visualisation solutions, rather than explaining their problems. Potentially they may act as a useful bridge in the knowledge gap, presenting a collection of domain independent tasks from which concrete domain tasks can be derived. Finally, using a task classification in this way may help with task specification at a consistent level of granularity and abstraction.

Note that there are relatively few documented examples of task classifications being used at this stage in the design process. One example is Ahn et al. [41], who demonstrate how their task design space could be used to help in the discovery of new tasks i.e. those tasks that analysts had not thought of during a requirements gathering process.

2.3.4.2 Data/operation abstraction design

Once concrete, domain specific tasks have been captured, the data/operation abstraction stage requires that they be translated into the language of information visualisation. The resulting set of abstract tasks (operations) is used as the basis for selecting visual encodings at the encoding/interaction technique design stage. Task classifications can be utilised at this stage to describe domain tasks in appropriate abstract terms [43], for example, Brehmer et al. [53] propose using their classification as a "lexicon for coding" observed tasks. The process of abstraction reveals similarities between tasks that may initially appear to be rather different [54]. This allows them to be meaningfully grouped together, thus categories of frequently occurring tasks can be identified. This can be useful when determining which tools to include when developing a system at the next stage of the design process.

2.3.4.3 Encoding/interaction technique design

Visual technique classifications can be of assistance in revealing the range of potential design solutions at the encoding/interaction technique design stage. They can be divided into those that categorise the range of visual representations of the data, and those that deal with interaction [45]. They may classify techniques according to algorithms used [55], data structure (e.g. [56]–[58]), similarity of encodings (e.g. [38], [39]), or task support (either visual operations e.g. [13], [59], or user intention e.g. [8], [60]). However, where task support is not the basis on which the classification is made, a mapping between the technique categories and the tasks which they support is required for the classifications to be of assistance in directly helping designers to select appropriate tools for inclusion in their systems. Wehrend and Lewis [8], were among the first to propose a "problem-oriented approach" to tool classification, categorising scientific visualisation techniques according to the sub-problems (tasks) and objects supported. This results in a task-technique "catalogue", which designers can use to look up potential visual solutions according to the problems for which they are trying to design. A particular advantage of the catalogue approach is that it provides a way to share visual solutions to similar problems across disparate application domains. Developing such task-technique mappings were also thought to be the first step in automated system design [8], [61].

Direct mappings between tasks and techniques for use in tool selection, however, may not always be possible or appropriate. Rind et al. [43] note the use of guidelines translating between *abstract* objectives (analytical tasks) and *abstract* actions (the means by which the objectives can be achieved), citing Andrienko and Andrienko [5] who - given the intentionally generic nature of their task framework - derive a set of general principles which can be utilised when designing exploratory tools. Roth [62] also notes the use of task classifications in the generation of design guidelines.

The importance of including process and provenance functionality in visual analysis systems, such as those described by [48], has recently been highlighted in the visualisation literature, and task classifications can play a role in identifying tasks for this purpose. Gotz and Zhou [63] and von Landesberger et al. [64] develop task classifications with this purpose in mind. Rind et al. [43] offer a more detailed discussion of the integration of tasks in visualisation artefacts.

2.3.5 Use in Evaluation

An often-cited motivation for developing task classifications is their use in the evaluation process. They can be of use when selecting representative tasks for use in experiments, acting as a “checklist” covering the range of possible tasks for inclusion, in a manner similar to that of the design process (Section 2.3.4). Brehmer et al. [53] outline potential uses of their characterisation of task sequences in four of Lam et al.’s [6] empirical evaluation scenarios: as a lexicon for coding observations when *understanding working practices* (as described in Section 2.3.4.3); to inform the design of experimental procedures when *evaluating user performance*; when specifying tasks for use when *evaluating user experience*, either as instructions in experiments or when constructing questionnaires and interview questions relating to user experience; and when coding observed behaviour *when evaluating visual data analysis and reasoning*.

A primary use of task classifications for evaluation purposes found in the literature is their use in characterising systems in terms of task support. This allows evaluators to assess individual systems in terms of their capabilities and limitations e.g. [44], or make comparisons across multiple systems e.g. [40], [44]–[46], [65].

2.4 **Task classifications in the visualisation literature**

2.4.1 General task classifications

Several general task classifications exist in the visualisation literature, including Shneiderman’s task by data type taxonomy [56], Amar et al.’s taxonomy of low level tasks [66], and Andrienko’s formal task framework for EDA [5]. However, the use of the term ‘task’ in the visualisation literature is “deeply overloaded”, being used at multiple levels of abstraction and granularity [10]. Recent work has therefore focussed on unifying extant task taxonomies [45], [67], and “untangling the terminology” surrounding the use of the term task, with Rind et al. [68] distinguishing three conceptual dimensions along which tasks can be categorised:

- **Perspective**: whether the tasks are *objectives* (analytical questions asked of the data i.e. *why* the task is carried out) or *actions* (discrete steps towards addressing the objectives i.e. *how* the task is carried out).

- **Abstraction**: a continuum from *concrete* tasks couched in application domain specific language, to *abstract* tasks which are described in a more generic manner.
- **Composition**: a continuum from tasks specified at a *high level* of composition to those broken down into specific *low level* subtasks.

They use these conceptual dimensions to categorise and compare abstract task classifications, with an additional distinction relating to abstraction: whether the classification is *generic*, or relates to a specific *data type*, *domain* or *tool architecture*.

As described in Section 2.3, abstract task classifications are necessary in order to be able to generalise beyond a specific use case, reuse methods, and facilitate communication amongst researchers. However, Brehmer and Munzner [67] recognise that there is still a need for data specific task classifications. Such classifications are particularly useful when assessing tools and techniques in terms of the tasks which they are able to support.

2.4.2 Static graph task classifications

Lee et al.’s taxonomy of static graph tasks [40] is a good example of a data-specific classification. They extend Amar et al.’s [66] low level task taxonomy for use with graphs, and categorise the resulting tasks into five groups: topology based; attribute-based; browsing; overview; and high level tasks. While they include in their high-level task category the general question, *how has the graph changed over time*?, they do not elaborate on the sub tasks involved.

Shneiderman and Aris [69] identify six challenges for network visualisations (basic networks; node labels; link labels; directed networks; node attributes; link attributes) and a number of associated high priority tasks, such as counting nodes and links; finding structural metrics; and structural and attribute based tasks similar to those described by Lee et al. [40]. However, they describe the number of potential tasks which a person may wish to carry out on a network as “unlimited”. They do not consider temporal graph tasks.

2.4.3 Temporal graph task classifications

With regard to temporal graph task taxonomies, Yi et al. [70] categorise visual tasks in temporal social network analysis by the level at which temporal change in the network can be analysed: nodal and dyad level (node or edge attributes, and associations between attributes), subgroup (based on connectivity or node attributes), and global level. They identify the general aspects of interest at each of these levels in relation to network evolution:

- At the nodal and dyad level, the emergence, growth and dissolution of nodes and ties.
- At the subgroup level, the processes of subgroup formation.
- At the global level, global changes in the network's topology over time.

They also note the importance of considering the relationship between attributes and graph structure. However, their tasks are expressed at a high level of composition, and they do not specify in detail the subtasks involved in the analysis of temporal change at each level.

Bach et al. [71] recently adapted Peuquet's [72] geo-temporal task framework for use with temporal graph data. The original framework consists of three dimensions, *when*, *where*, and *what*. Bach et al. redefine the *where* and *what* dimensions to capture the lack of fixed spatial positions in temporal graph data. Tasks are formulated based on two known dimension values, with the third dimension's value to be found, giving three general task types (what + when = where; when + where = what; where + what = when; items to the left of the equation are specified, items to the right are those which are to be found). This approach is similar to that of the Andrienko framework [5] (discussed in Chapter 3), with the task types being comparable to the Andrienko 'lookup' task category. However, the Andrienko approach is more detailed, and captures higher level tasks such as comparison and finding correlations and dependencies, not specified in this framework.

The most detailed example of a temporal graph task taxonomy is Ahn et al.'s taxonomy for network evolution analysis [41]. The taxonomy consists of three dimensions: Entity, Property, and Temporal Feature. 'Entity' follows Yi et al. [70] in their distinction of levels of analysis; ‘Property’ distinguishes between structural attributes and domain properties. These dimensions capture *what* should be observed. ‘Temporal Features’ explain *how* these items should be analysed: as ‘Individual events’ at single time points or ‘Aggregate events’ over a period of time. The three dimensions are used to construct a task design space, which they use as a basis for characterising existing visual techniques by the tasks which they are able to support. The limitations of Ahn et al.’s taxonomy are discussed in detail in Section 6.2.1.

2.5 **Tasks in the graph and temporal graph visualisation literature**

In addition to task taxonomies, there are two additional sources of temporal graph tasks in the literature: the tasks which systems and techniques have been designed to support, and the tasks employed in studies evaluating visualisation techniques.

2.5.1 Tasks in systems and techniques papers

With some notable exceptions e.g. [69], [70], [73], the discussions of tasks in the systems and techniques papers are surprisingly limited, and often couched in domain specific terms. For example, Erten et al.’s [74] tasks relate specifically to the the ACM digital library’s co-authorship network: *What were the hottest topics in computing in the 1990’s? Which research areas are experiencing steady decline/rapid increase? Which research communities are open and well-connected?* Meanwhile, Gloor and Zhao’s [75] tasks are phrased in terms of their interest in communication technologies: *Do social networks depend on the interaction technology? Does the same group of people exhibit different network attributes when interacting via telephone, email, face-to-face or other?* Additionally, systems are often specific in their focus, thus tasks are constrained to a particular subset e.g. Kang et al. [76] focus on the specific task of analysing change in group membership of a pair of individuals over time. As discussed in Section 2.4.1, the term “task” is used ambiguously in the visualisation literature in general, and this is reflected in the tasks described in the systems and technique papers, which are specified at varying levels of perspective, abstraction, and composition.

Evaluative studies tend to employ only a limited number of the range of possible tasks. The tasks in Ghoniem et al.'s [77] study of static graph representations consider the basic characteristics of vertices, paths, and subgraphs. They formulate seven generic tasks concerned with gaining an overview of a graph's structure, including estimating the number of nodes and edges in the graph, and finding particular nodes, links, and paths. They do not consider tasks involving node or edge attributes.

Purchase and Samra [20] and Archambault et al. [27] use five tasks in their studies investigating mental map preservation; similar (edge-based) questions were included in [17]. They consider:

- global and local structures.
- the evolution of node degrees
- node and edge appearance/endurance
- growth in number of nodes, and
- the readability of paths over time.

Unlike other studies, tasks were purposefully presented in a domain context, to make them more understandable to study participants. Little explanation is given in these studies as to how or why the tasks used were chosen. Again, tasks involving attributes are not considered.

When evaluating techniques for analysis of graphs with associated time-series data, Saraiya et al. [73] formulate eleven tasks for use in their evaluation. While the tasks are based on those commonly employed in the bioinformatics domain, they were purposefully abstracted for use in the study, as study participants did not necessarily have domain knowledge. The tasks focus on timeseries associated with node attribute values in the graph context, and are categorised according to the number of time points involved.

Farrugia and Quigley [78] provide a comprehensive discussion of the tasks used in their study. They distinguish four task categories based on a combination of level of analysis (global network overview vs local individual node level) with temporal search space (specified vs unspecified time period), in conjunction with the static graph tasks of Lee et al. [40], a selection of which they formulated for the dynamic context. However, they give only a few examples of the tasks used.

2.6 Taxonomies of visual techniques

The general ways in which techniques in the visualisation literature have been categorised were discussed in Section 2.3.4.3. Classifications of techniques are further discussed in Chapter 7. Visual technique taxonomies can be of assistance in revealing the range of potential design solutions. They can be divided into those that categorise the range of visual representations of the data, and those that deal with interaction [45]. They may classify techniques according to algorithms used [55], data structure (e.g. [56]–[58]), similarity of encodings (e.g. [38], [39]), or task support (either visual operations e.g. [13], [59], or user intention e.g. [8], [60]). In the graph case, von Landesberger et al. [79] classify graphs according to their time dependence and graph structure (Figure 2), and use this classification as the basis for their discussion of visual techniques.

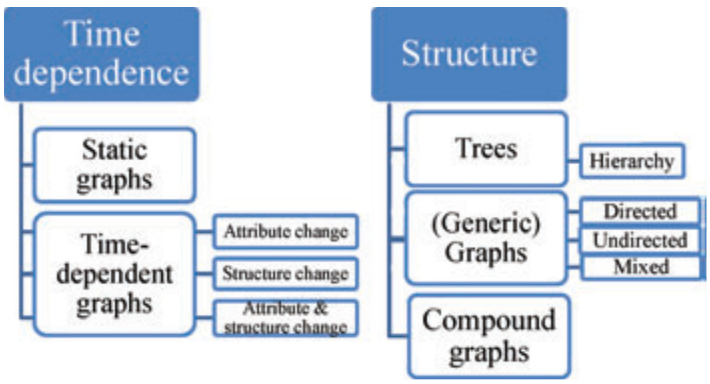


Figure 2 classification of graphs by time dependence and structure (von Landesberger et al., [79]**, Figure 3).**

For general graphs, Schulz and Schumann [80] distinguish four dimensions upon which network visualisation techniques can be categorised: dimensionality (2D or 3D representation), directionality (directed or undirected); edge representation (explicit vs. implicit); node layout (free, styled, or fixed). For trees, Graham and Kennedy [81] identified five basic types of tree representation: node-link, nested, adjacency, indented list, and matrix (Figure 3). Schulz et al. [38] surveyed implicit hierarchy techniques and extracted four dimensions (dimensionality, edge representation, node representation, and layout) which they use to construct a design space of possible representations.

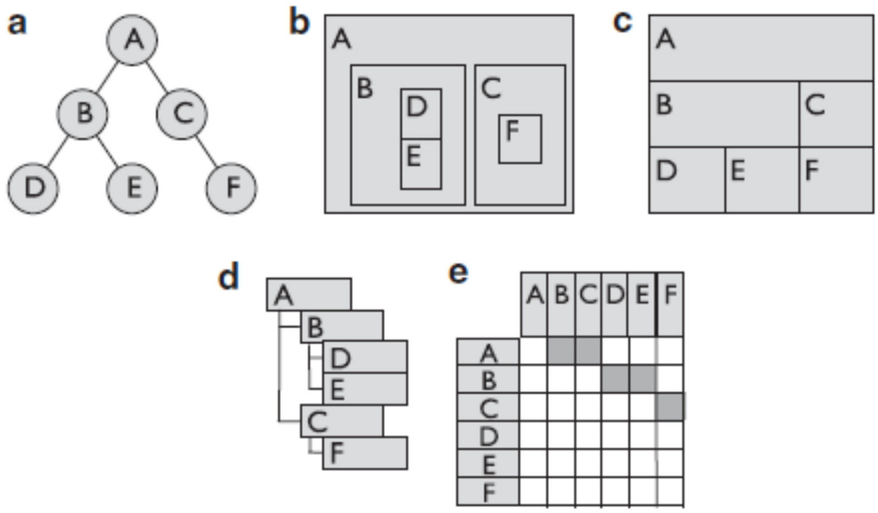


Figure 3 Five basic types of tree representation (Graham & Kennedy, [81]**, Figure 3): (a) node-link, (b) nested, (c) adjacency, (d) indented list, (e) matrix.**

For temporal graphs, Hadlak et al. [82] classify large dynamic graphs according to the reduction techniques involved: whether the temporal or structural element of the graph is reduced, and whether the reduction is via abstraction or selection, or is unreduced. They combine these dimensions to produce a design space of temporal graph representations (Figure 4).

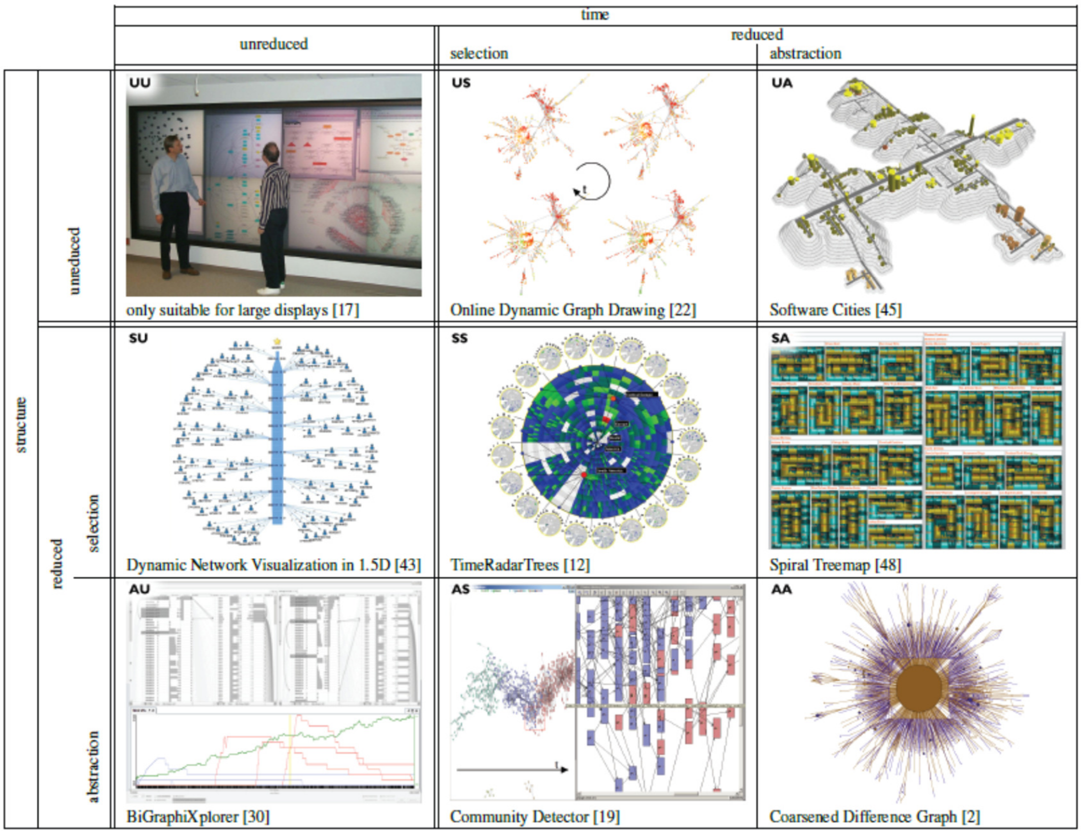


Figure 4 Hadlak et al.'s categorisation of visual approaches for large dynamic graphs ([82]**, Table 1).**

Both Federico et al. [28] and Rufiange and McGuffin [83] categorise temporal graphs with respect to the temporal encoding used.

Recently, Beck et al. [84] classified existing methods for representing dynamic graphs, firstly according to the temporal encoding used: either animated views or static timeline representations. Animated views are subdivided by the layout algorithm employed; timeline representations are subdivided according to the graph structural encoding used (node-link or matrix), with further subcategories considered for each structural encoding (Figure 5). They also map published techniques to the categories of their taxonomy. As their classification is based on existing techniques, the mapping is not intended to reveal unexplored possibilities, but shows which techniques are most commonly employed, and the areas in which less work has been undertaken.

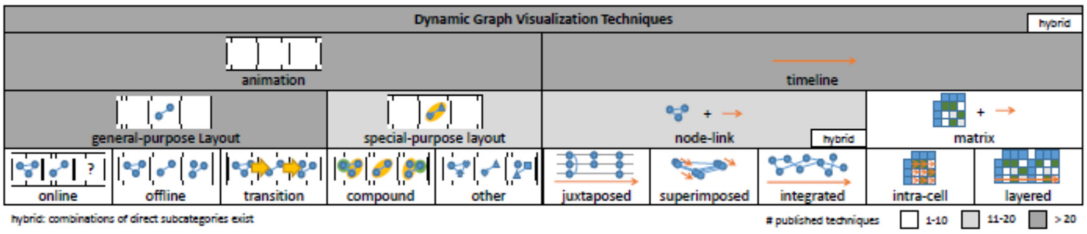


Figure 5 Beck et al.'s classification of dynamic graph visualisation techniques ([84], Figure 1). The number of publications employing a particular technique is indicated using cell colour.

2.7 Mappings between temporal graph tasks and visualisation techniques

As mentioned in Section 2.4.3, Ahn et al. [41] use the dimensions of their temporal graph task taxonomy to construct a design space, to which they map 53 existing visualisation systems according to the tasks which they support. Based on this mapping, they identify tasks which are well supported, and those which are not.

Beck et al. [85] suggest matching dynamic graph visualisation tools to application requirements through the use of profiles. Their methodology involves generating two types of profile: a visualisation profile, which rates the tool in terms of its support for the set of aesthetic criteria for dynamic graph visualisation; and an application profile, which estimates the relative importance of each of the aesthetic criteria based on the characteristics of the dataset and the required tasks. Having translated the visualisation strengths and application requirements into a common set of values, the best fit between the two profiles can be found. However, at the time of writing, the task domain and technique domain had not yet been fully explored, specified, and categorised, therefore their methodology required expert knowledge of the possible range of available techniques and the tasks which may be required when exploring dynamic graph data. The work in this thesis could help addresses these deficiencies by providing two useful pieces required to implement their methodology: a classification of the available tools and techniques, and a comprehensive classification of the task domain.

2.8 Discussion

So far this chapter has outlined the important role that taxonomies can play in the design and evaluation of visualisation systems and techniques, and how they can be useful tools for directing attention to potential avenues for future research (Section

2.3). Having reviewed the existing work which has been carried out in developing task and technique taxonomies for temporal graphs (Sections 2.4.3 and 2.6), let us return to the questions outlined in Chapter 1 and consider whether these are sufficiently addressed within the existing literature.

To date, three task taxonomies have been developed specifically for temporal graphs: Bach et al.'s [71] adaption of Peuquet's [72] geo-temporal task framework for use with temporal graph data; Yi et al.'s [70] categorisation of visual tasks for temporal social network analysis; and Ahn et al.'s taxonomy for network evolution analysis [41]. However, as discussed in Section 2.4.3 and demonstrated in the evaluation of Section 6.2.1, all of these taxonomies have shortcomings in terms of task coverage. Tasks in the temporal graph literature (considered in Section 2.5) are specified at varying levels of granularity and abstraction, while evaluative studies tend to include only a small number of tasks. Collating these tasks into a single taxonomy would likely prove difficult due to the different levels at which tasks are specified, and would in no way ensure task coverage, given the surprisingly limited consideration given to tasks in the literature. Therefore, research question 1, *what are the possible exploratory analysis tasks that temporal graph visualisation might need to support?* has not yet been fully addressed by the existing work.

As discussed in Section 2.6, a number of different approaches have been taken to categorising both static and temporal graph techniques. Temporal graph techniques have been classified according to the reduction techniques employed (in the case of large graphs) and also the temporal encodings used. All of the classifications to date have considered only existing techniques. The first part of research question 2, *which visual techniques, tools, and approaches, have been developed to support exploration of temporal graph data?* has therefore been partially considered by the existing literature, however the categories identified in the existing classifications would benefit from being brought together into a more unified classification. As all of the classifications to date have considered only existing techniques, the second part of research question 2, *are there any unexplored opportunities for visual techniques?* has not yet been answered, and there is room for exploration of the possibilities for new techniques.

In terms of the relationship between techniques and the tasks which they support, as discussed in Section 2.7, Ahn et al. [41] map existing techniques from the literature to their task taxonomy. However, as outlined in Section 2.4.3, the task coverage of their taxonomy is incomplete. Further research is therefore required to fully address research question 3, *which visual techniques support which tasks?*

Finally, research question 4, *for the tasks identified in (1), are there suitable visual techniques or are new/better techniques required?,* cannot be answered until questions 1 and 3 have been fully addressed, therefore it remains an open research question.

Having established the limitations of existing work in tackling the four research questions outlined in the introductory chapter, this thesis seeks to address these questions by:

(1) developing a taxonomy and design space of temporal graph tasks, which will provide more complete task coverage than the existing works.

(2) exploring the 'space of the possible' of temporal graph visualisation techniques, through construction of a design space to which existing techniques from the literature are mapped, revealing currently unexplored techniques.

(3) considering the techniques which can support each of the task categories of the taxonomy.

(4) identifying tasks which could benefit from new or better techniques for their support.

2.9 Summary

This chapter has explored the motivation for the development and use of taxonomies in visualisation research, and reviewed the existing work relating to task and visualisation technique taxonomies for temporal graphs. The existing work has been considered in relation to the research questions proposed at the beginning of this thesis; this has established the extent to which the questions have already been addressed by existing work, and demonstrates that further work is required to fully answer these questions, serving as both justification and motivation for the work in this thesis.

Chapter 3 Task Taxonomy: Methodology

This chapter reviews the possible approaches to task classification construction, threats to validity at each stage of development, and methods by which these threats can be mitigated. The task taxonomy for temporal graph exploration presented in this thesis is based on the work of Andrienko and Andrienko [5]. The reasons for its selection are discussed, and an overview of the formal approach it employs is given in this chapter, along with a discussion of the limitations of the framework in the graph case, which necessitated the extension to the taxonomy outlined in Chapter 4, and allowed its application to the temporal graph case outlined in Chapter 5.

3.1 Constructing task classifications

While recent work has been carried out in establishing appropriate design processes when developing visualisation systems and technique e.g. [10], [12], [50], [54], and despite the increasing number of task classifications being developed, very little attention has so far been given to the process of classification construction. In order to investigate the construction methods employed when developing a task classification, a review of the literature was carried out. Rind et al.'s [43] list of 31 abstract task categorisations was used as the basis for this review. 26 of these classifications are discussed below. Three of the publications, [86]–[88], were unavailable; [54] and [67] are considered together, as the discussion relating to the construction and evaluation of this classification is presented in [67]; [89] was excluded as it is the published version of the task classification developed in this thesis. The list of publications included in the review along with a summary of the construction approaches and evaluation methods employed by each work is included in Table 1. Reference was also made to literature relating to the visualisation design process and evaluation practices, where appropriate.

Classification	Terminology (used in paper)	Task Generation	Categorisation	Description	Evaluation
Ahn et al. [41]: network evolution	Taxonomy, design space	Derived from literature – existing systems/techniques Expert review	-	Verbal	Properties: descriptive power; usability Usage: design process
Alsallakh et al. [65] set-typed data	Classification	Derived from literature – existing systems/techniques	-	Verbal	Usage: evaluation
Amar and Stasko [46]: prototypical analysis tasks	Examples of common tasks	Not specified	-	Verbal	Construction method Property: descriptive power Usage: design process; evaluation
Amar et al. [66]: low-level visualisation tasks	Taxonomy	Survey of visualisation experts	affinity diagramming	Verbal	Property: comprehensiveness
Andrienko and Andrienko [5]: exploratory data analysis	Typology	Formal modelling approach	n/a	Functional; Verbal	Properties: comprehensiveness; real world nature of tasks
Brehmer and Munzner [67]; Munzner [54]: abstract visualisation tasks	Typology	Extant classifications – unifies; influenced by/derived from Author's own knowledge	-	Verbal; Faceted	Properties: descriptive power; syncretism
Brehmer et al. [53]: task sequences for dimensionally-reduced data	Characterisation of task sequences	Interviews with domain experts	iterative coding process	Verbal	Construction method Property: comprehensiveness
Chuah and Roth [61]: interaction	Framework	Extant classifications - influenced by/derived from Derived from literature – existing systems/techniques	-	Verbal	-

Classification	Terminology (used in paper)	Task Generation	Categorisation	Description	Evaluation
Gotz and Zhou [63]: insight provenance	Catalogue, taxonomy	Extant classifications - influenced by/derived from Derived from literature – existing systems/techniques; user studies Observation of visualisation users	-	Verbal	Usage: design process
Heer and Shneiderman [48]: interactive dynamics	Taxonomy	Not specified	-	Verbal	-
Lammarsch et al. [90]: time-oriented data	Rule set	Extant classifications – extends [AA06] Formal modelling approach	n/a	Functional; Verbal	-
Lee et al. [40]: graphs	Taxonomy	Extant classifications - influenced by/derived from Derived from literature – evaluation tasks	-	Verbal	Usage: evaluation
Liu and Stasko [91]: mental models	Categorisation	Extant classifications - influenced by/derived from	-	Verbal	-
Meyer et al. [92]: comparative genomics	Characterisation, taxonomy, design space	Derived from literature – existing systems/techniques Interviews with domain experts	-	Verbal	-
Pretorius et al. [93]: multivariate networks	Framework	Extant classifications - influenced by/derived from	-	Verbal	-
Rind et al. [94]: electronic health records	Classification	Extant classifications – extends [YKSJ07]	-	Verbal	-
Roth [62]: interactive cartography and geovisualisation	Taxonomy	Interviews with domain experts	card sorting with domain experts	Verbal	Construction method

Classification	Terminology (used in paper)	Task Generation	Categorisation	Description	Evaluation
Sacha et al. [44]: knowledge generation, visual analytics	Model	Extant classifications - unifies	-	Verbal	Properties: descriptive power; syncretism Usage: evaluation
Schulz et al. [45]: visualisation tasks	Design Space	Extant classifications – unifies; influenced by/derived from	-	Verbal; Faceted	Properties: descriptive power; comprehensiveness Usage: design process; evaluation
Sedig and Parsons [47]: action patterns	Theoretical framework, Catalogue	Extant classifications - influenced by/derived from Derived from literature – existing systems/techniques	Identify common characteristics and uses; use of abstraction	Verbal	Property: real world nature of tasks
Shneiderman [56]: visualisation tasks by data type	Taxonomy	Not specified	-	Verbal	-
Suo [95]: network security	Taxonomy, design space	Extant classifications - influenced by/derived from Author's own knowledge	-	Verbal	-
Valiati et al. [7]: multidimensional data	Taxonomy	Extant classifications - influenced by/derived from Observation of visualisation users	-	Verbal	Property: descriptive power
von Landesberger et al. [64]: interaction	Taxonomy	Extant classifications – unifies	-	Verbal	Property: syncretism Usage: design process
Wehrend and Lewis [8]: scientific visualisation	Classification/ catalogue	Derived from literature – problems addressed	-	Verbal	Construction method Property: comprehensiveness Usage: design process

Classification	Terminology (used in paper)	Task Generation	Categorisation	Description	Evaluation
Yi et al. [13]: interaction	Categorisation	Extant classifications - influenced by/derived from Derived from literature – existing systems/techniq ues; problems addressed Review of commercial systems	affinity diagramming	Verbal	-

Table 1 Summary of task classifications reviewed, including terminology used to describe the classification, and the construction approaches and evaluations reported in these papers. The original list of publications is based on that used in Rind et al.'s [43] review of the task design space.

There are a number of ways in which classifications can be constructed, although little reflection on the processes involved is to be found in the visualisation literature. Schulz et al. [SNHS13] consider the process of establishing recurring visualisation tasks and their description. They also discuss the consolidation of existing works. When taking a taxonomic approach to classification structure (i.e. where a set of existing items are gathered and grouped together based on their similarity), three main steps can be identified: (1) generate the tasks, (2) collate and order them, (3) describe them. In contrast, what shall here be referred to as *conceptual approaches* to classification construction—such as typologies and design spaces, as outlined in Section 2.3.1—begin with a set of important characteristics upon which tasks can be distinguished. In this case, rather than gathering a set of tasks, a set of conceptual dimensions are identified and used as the basis of classification construction. While these may need to be ordered in some way, the same process of rationalisation of tasks into categories required by step (2) of the taxonomic approach is not necessary. Some form of description of the resultant categories is still required, although for design spaces, the combination of choices along each dimension often serves to suitably define the category.

Different threats to validity arise from the different approaches that can be taken at each stage of classification construction, which consequently require different approaches to validation. The following sections discuss the approaches to construction, threats to validity, and possible means of mitigating these threats at each stage of the construction process. The discussion is largely structured around the three steps in taxonomic construction and each step is summarised in Tables Table 2Table 3Table 4Table 5. A discussion specific to conceptual approaches is included towards the end of this section, which is summarised in Table 5.

3.1.1 Task Generation

Task generation refers to the process of obtaining a set of tasks upon which a classification is based. Such a definition is most fitting when applied to taxonomies, where a set of items are collected and then organised. Here the idea is expanded to include the process by which the dimensions of other forms of classification (such as typologies and design spaces) are obtained. Schulz et al. [45] describe a number of common approaches to obtaining recurring visualisation tasks, including surveying individuals, observing visualisation users, and inferring from existing visualisation systems. In reviewing the literature, the most prevalent approach to task generation was found to involve literature based methods (20 of 26 classifications): either involving extant classifications (16 of 26) or deriving tasks from the literature (9 of 26). Extant classifications may be unified [44], [45], [64], [67]; extended (e.g. to a specific data type ([5] by [49]), or for domain specific purposes ([13] by [94])); or – most commonly - used to derive, or cited as influencing, the task categories [7], [13], [40], [45], [47], [63], [93], [95], with a small number making use of theoretical works from across a wider range of disciplines, such as HCI and cognitive science e.g. [47], [63], [67], [91]. Tasks can be derived from the literature by reviewing: existing systems/techniques for the tasks which they support [13], [41], [47], [61], [63], [65], [92]; problems addressed in the literature [8], [13]; tasks utilised in user evaluations [40]; or studies examining users' visual analytic behaviour [63]. Far less common are the use of empirical methods to elicit tasks (8 of 26 classifications), including interviews with domain experts (either in a single domain [62], [92], or across multiple domains [53]); surveys of those familiar with visualisation [66]; observational studies of people using visualisation systems [7], [63]; reviews of commercial systems [13]; and expert reviews of the resultant classification to find missing tasks [41]. An alternative to literature based and empirical methods is the use of theoretical approaches, where a formal modelling approach is taken [5]. Finally, authors frequently draw on their existing knowledge of literature, systems, and practices when constructing classifications. Two of the papers reviewed explicitly acknowledge this (drawing on “new thinking ”[67], or the author’s experience [95]), but many more likely do this implicitly, including the three papers which did not specify the means by which their tasks were generated [46], [56], [59]: especially likely given their inclusion of extensive reference to the literature.

3.1.1.1 Task Generation: threats to validity

The two main threats at this stage are (1) gathering the wrong tasks (2) gathering an incomplete set of task. These threats arise in different ways depending on the method used.

The threats to validity when gathering tasks reported in the literature or via empirical methods such as interviews and observations are the same threats encountered at the task gathering stage of the design process, and thus the problems are well documented. Relying on tasks reported in the literature requires a certain level of understanding of domain terminology on the classification constructor’s part (who is likely to be a visualisation researcher), and/or a similar problem having already been tackled in the visualisation literature (which preferably would include a clear characterisation of the problem; however, as noted by Munzner [10] problem characterisation papers are somewhat lacking in the visualisation literature). Talking to domain experts has a number of difficulties associated with it. In practical terms, access to domain experts may be limited in terms of the their availability [96]. Relying on experts from a single domain may also skew the set of tasks towards that of the represented domain [89], a problem if the resultant classification is intended for more generic use. A more general, well-known issue in HCI and psychology is that people find it difficult to accurately introspect about their needs and articulate them [10], [12]. This difficulty is compounded when developing task classifications for Exploratory Data Analysis, where the purpose is exploration, and the potential analytical tasks involved in the exploration are not necessarily known at outset [4], [5]. Sedlmair et al. [12] also note the need to keep discussions focused on tasks; one pitfall they identify is allowing experts to focus on possible visualisation solutions, rather than explaining their problems. The gap in understanding the terminology used by domain experts, may also be a factor during such discussions.

Relying on tasks generated by those familiar with visualisation, as opposed to domain experts (as per the strategy of Amar et al. [66] who surveyed visualisation students to generate a set of analysis tasks) may also result in wrong or missing tasks. Indeed Amar et al. reflect on whether they would have obtained a different set of tasks had they surveyed professional analysts.

Observational strategies require that working methods be observed e.g. observing the domain expert using an existing visualisation system or some other tool; however, this requires that at least some method for tackling the problem already exists, which may not be the case for novel problems. Where systems do exist, researchers must still be careful to establish that the problem being tackled is indeed the right one. Moreover, the inherent lack of access to the internal mental processes of participants during fly-on-the-wall observation techniques makes observing the cognitive tasks which they are performing difficult, (although contextual enquiry, where the researcher interrupts to ask questions during the observation, may overcome some of this difficulty) [12].

Adopting a multi-strand approach to task gathering may be one way to reduce the chances of gathering the wrong or an incomplete set of tasks upon which to base a task classification. Downstream evaluation of the resulting classification (using approaches such as those outlined in Section 6.1) may also highlight problems at this stage.

Finally, the principle of ‘garbage in, garbage out’ applies where the categories of extant classifications are used as the basis for constructing a classification: the validity of the resultant classification will be affected by the methods involved in constructing the original classifications. Those which have not been validated during construction or evaluated in a final form may contain errors which could be propagated to future classifications. Downstream evaluation of the resultant classification is therefore necessary where extant classifications are the basis for construction.

Method	Threat	Mitigation
	General threat -wrong or missing tasks, arising from:	Multi-strand approach to task gathering Downstream evaluation
Derive from literature (existing systems/techniques; existing problems; tasks in evaluations)	- Understanding domain terminology - Requires similar problem having already been tackled	
Interviews with domain experts	- Expert's availability - Skewing tasks towards a single domain - Difficulties with introspection/in articulating tasks - Maintaining focus on task discussions	
Surveys of visualisation experts	Wrong people	
Observational strategies	- Method of tackling problem must already exists - Lack of access to internal mental processes	
System reviews	Method of tackling problem must already exists	
Author's own knowledge	- Missing tasks - skew towards particular domain	
Derive from extant frameworks	Validity of original classification used	Downstream evaluation

Table 2 Task generation: summary of methods, threats to validity, and approaches for mitigating threats.

3.1.2 Categorisation

In the taxonomic approach, once tasks are gathered, some method of establishing meaningful categories is required. In 13 of the 26 classifications we reviewed, a set of tasks were gathered either from the literature or through empirical means. Of these, only five reported the method of categorisation employed when grouping the tasks. These included identifying common characteristics and uses of techniques and abstracting beyond the details of particular implementations [47]; an iterative coding process [53]; affinity diagramming [13], [36]; and card sorting with domain experts [62].

3.1.2.1 Categorisation: threats to validity

The two main threats at this stage of classification construction are incorrect and missing task categories. These threats arise from two directions: upstream, from the set of items collected at the task generation stage, and at the current stage, from the method by which categorisation is performed.

In terms of upstream threats, where a taxonomic process is being followed, the threat to validity may be propagated from the task generation stage, i.e. where the wrong or an incomplete set of tasks is collected, categories based on these items are likely to be flawed. Similarly, where classifications are constructed from the categories of extant classifications which have not been evaluated, any issues with the original classifications will potentially be propagated to the classification being developed. These threats may be mitigated by carrying out validation at the task generation stage, or identified during downstream evaluation of the final classification.

In terms of threats arising at the construction stage, when carrying out a taxonomic procedure, determining what constitutes 'similarity' between tasks can be a non-trivial problem, particularly where tasks are drawn from across a range of application domains and may be specified inconsistently (i.e. with respect to Rind et al.'s [43] distinctions: at different levels of composition, abstraction and even in terms of actions vs objectives). Reasoning about similarities and differences between tasks often requires some level of abstraction. As Munzner notes when discussing abstraction in the visualisation design process, apparent differences between tasks are often misleading as "*…there are a lot of similarities in what people want to do once you strip away the surface language differences*" [54] p 43-44. The use of systematic methods such iterative coding, affinity diagramming, and card sorting techniques (as outlined in the studies mentioned above) are one way to mitigate against producing the wrong categories. Some of the evaluative methods identified by McKenna et al. [50] for the 'understand' activity of their design activity framework could also be of potential use at this stage in the construction process. However, consideration also needs to be given to *who* is carrying out these processes. In most cases, categorisation was performed by the classification constructors (normally visualisation researchers). While this may be a valid approach (often the intended users of the resultant classification are visualisation researchers), reasoning about similarities and differences amongst domain tasks may best be performed by domain experts

One further threat to the potential usefulness of a classification is its structure, in terms of the granularity (size of categories), complexity, and depth (levels in a hierarchy) of categories. While discussion of these aspects was not covered in the literature reviewed, they have been discussed in other disciplines which develop classifications (such as biology and information management). In terms of granularity, use of wide categories may have the advantage of producing a simpler classification with fewer categories, but may group together tasks with important distinctions (for example, where the classification is intended for use in a task-technique mapping, grouping tasks widely may result in difficulty in finding techniques supporting the full range of tasks). During the task categorisation processes, subtle yet important distinctions between tasks may be lost, and less commonly occurring, but important, 'corner case' tasks may be discarded. Meanwhile, narrow categories can result in the opposite problem – creating an overly complex structure by differentiating sets of tasks which could meaningfully be grouped together. Similarly, classifications which employ a hierarchical structure may wish to consider the depth and complexity of their structure. While other research areas have developed rules (such as ensuring consistency in depth to promote a 'balance' to the hierarchy, easing predictability when browsing and navigating the structure [97], or limiting the depth of the hierarchy, as in the (now outdated) '3 click rule' for web navigation) the potential effect of hierarchical structure has not been considered when developing task classifications. The optimal structure of a classification will likely depend very much on individual circumstances and intended use. Downstream validation of the resultant classification for the intended purpose and with the intended group of end users (such as that performed by Ahn et al., [41], who evaluated their classification via interviews with a number of experts from different domains) is therefore important.

Finally, where extant categories are combined to either unite, or improve upon existing classifications, it is important to validate that this has been achieved. In the former case this may be done by demonstrating that all categories have been subsumed by the new classification (e.g. Brehmer and Munzner [67] map all extant categories to the categories of their classifications, while Sacha et al. [44] use discussion and illustration to demonstrate how the extant categories have been incorporated into their framework.) In the latter case, a discussion of the shortcomings of extant works and necessary additions helps validate the need for the new classification. Where additional categories are identified, validating the processes involved in their identification may require use of the methods discussed in this section and also at the task generation stage.

Method	Threat	Mitigation
Ad hoc methods	Structural issues (granularity and depth) Wrong or missing categories arising from: - Upstream threats (wrong or missing tasks at task generation stage; validity of extant classifications used) - Inconsistently specified tasks - Wrong people performing categorisation	Upstream validation Downstream evaluation Use of systematic methods
Systematic methods (iterative coding; affinity diagramming; card sorting with domain experts)		

Table 3 Categorisation - summary of methods, threats to validity and associated approaches to threat mitigation

3.1.3 Category Description

Schulz et al. [45] identify four ways in which visualisation tasks can be described: verbal, functional, logical, or faceted. They also note that task descriptions may be hierarchical, allowing larger tasks to be represented as sequences of smaller subtasks. Almost all of the task descriptions used in the classifications surveyed were verbal. Some, such as Brehmer and Munzner's [67] typology and Schulz's task design space [45], describe tasks in a faceted manner, in which case the task description is composed of a series of elementary components. Only the work of Andrienko and Andrienko [5] and Lammarsch et al. [49] (whose work extends it), provide a functional notation.

3.1.3.1 Category Description: threats to validity

The main threats at this stage of classification construction are ambiguous or unclear descriptions, and descriptions specified in an inconsistent manner.

Ambiguous or unclear descriptions are a problem when the classification is intended for adoption by others. While the use of formal notation avoids ambiguity and allows highly nuanced distinctions between tasks to be made, it has the disadvantage that it may be difficult for those unfamiliar with the notation to read and understand. Inclusion of verbal descriptions alongside formal notation helps overcome this limitation. In a minority of cases, it was noted that some of the verbal descriptions found in the review were too brief to fully grasp the intended meaning of the category, being only a few words long. The format used by Yi et al. [13], which employs verbal descriptions and examples, is an example of good practice (emphasis added): *"To each category, as a title, we assigned a **short identifying name** (e.g., Select) and also an **illustrative phrase** that captures the essence of the user's intent in performing the interaction. We **describe each category to provide a definition** of what it means and we also include **exemplary individual interaction techniques** that fall within that category."*

The problem of overloading of the term "task" (as discussed in Section 2.4.1) is evident when describing task categories, in that descriptions are not always specified in a consistent manner. While some of this may stem from earlier stages in the construction process (e.g. a number of the interaction classifications have been accused of conflating actions and objectives, which may arise at the task generation or categorisation stages), in order to describe each task category in a consistent manner, it is useful to keep in mind Rind et al.'s [43] distinctions between actions and objections, and the varying levels of abstraction and composition, when constructing task descriptions.

Method	Threat	Mitigation
Verbal	- Ambiguous/unclear descriptions - Descriptions too brief - Descriptions specified in an inconsistent manner	Describe category in sufficient detail e.g. Yi et al.'s format Provide concrete examples
Formal notation	Difficult for those unfamiliar with notation	Accompany with verbal description

Table 4 Description - summary of methods, threats to validity and associated approaches to threat mitigation

3.1.4 Conceptual Approaches

As described above, conceptual approaches begin with a set of conceptual dimensions upon which tasks can be distinguished, and result in a set of categories which represent concepts rather than empirical cases. Some means of establishing these dimensions is therefore required. As outlined in Section 3.1.1, dimensions are often gathered from extant classifications, for example, both Schulz et al.'s [45] design space and Brehmer and Munzner's typology [67] draw on previous work to identify the dimensions of their classifications. An alternative approach is the formal modelling process used by Andrienko and Andrienko [5] who manipulate a metaphorical mathematical function in order to identify the types of tasks specified by their task typology.

As the dimensions are established at outset, there is no need for the categorisation step of the taxonomic approach. However, where dimensions are gathered in a taxonomic fashion, some means of rationalising them and establishing which dimensions to include in the classification is required. Where categories are derived from extant classifications, these need to be combined to form the new system. It was found in the review that the process used to synthesise extant classifications is rarely reported, although how the resultant classification fits with those on which it is based is sometimes discussed and/or illustrated.

Some form of description of the resultant categories is still required, therefore the discussion in Section 3.1.3 is relevant. For design spaces, the combination of choices along each dimension often serve to suitably define the category.

3.1.4.1 Conceptual approaches: threats to validity

There are two main threats to validity for conceptual approaches: missing categories and reification.

While formal modelling approaches are able to claim completeness with respect to the model used [5], the classification is only as comprehensive as the model or dimensions upon which it is based. As discussed in Section 3.1.1, where extant categories are utilised, consideration needs to be given to their provenance. Downstream evaluation of the resulting classification using approaches such as those which are discussed in Section 6.1 with regard to comprehensiveness may highlight problems with missing or inappropriate categories.

Conceptual approaches also face a unique threat to validity: the question of whether the tasks are in fact 'real world', as opposed to constructs of the process employed. Bailey [30] refers to this as the problem of “reification”, where theoretical constructs that do not exist empirically are ‘reified’ and treated as ‘real’ empirical entities. Providing concrete examples goes someway to mitigating this threat, however, validating the real world nature of tasks is a tricky problem, which is discussed further in Section 6.1.3.3

Method	Threat	Mitigation
Gather and rationalise dimensions e.g. use extant categories	Wrong/missing dimensions Provenance of extant categories	Downstream evaluation
Functional modelling	Wrong/incomplete model	
Both approaches	Reification	Provide concrete examples

Table 5 Conceptual Methods - summary of methods, threats to validity and associated approaches to threat mitigation

3.2 Selected approach in this work

A primary intention of this work is to elucidate the tasks involved in exploring temporal graph data. As outlined in Chapter 2, (and discussed further in 6.2.1) all of the extant temporal graph task taxonomies have shortcomings, particularly in terms of task coverage. Having considered the possible approaches to developing a task classification, it was decided to adopt a formal approach and apply the Andrienko framework [5] to the temporal graph case in order to identify the tasks involved in exploring temporal graph data.

The Andrienko framework is well-respected in the visualisation community, using a systematic process to set out the possible tasks which may be encountered in an Exploratory Data Analysis scenario. Having survived 10 years of use 'in the wild' (see Section 6.1.5) it is believed to offer a solid basis upon which to derive a task classification specific to temporal graph data. The primary advantage of adopting this formal approach is in task coverage. Andrienko offer a formal proof to show completeness of their framework with respect to their chosen model. As the framework is intentionally domain independent, it is especially relevant to this work, which is particularly interested in exploring the 'space of the possible' across application domains and identifying potential areas for future research opportunities. Taking a formal approach mitigates many of the threats to validity discussed above when task gathering directly from domain experts or domain specific literature/extant systems, particularly when it would be necessary to carry out this process across multiple application domains. It also circumvents the difficulties noted relating to abstracting tasks, ensuring task specification at a consistent level of perspective, abstraction, and composition. The design space approach utilised in this work (discussed further in Section 5.4) also avoids loss of important corner-case tasks during categorisation. Constructing a task design space is one way in which the nuanced distinctions between tasks can be maintained, whilst showing meaningful high level categories. Not only does a task design space elucidate all possible permutations of tasks, such structures allow a 'slice and dice' approach to be taken to task categorisation. This is useful, as the multiple dimensions mean that all of the tasks will fall into more than one category

Finally, the use of formal notation to describe tasks avoids ambiguity and allows highly nuanced distinctions to be made. Coupling this with verbal descriptions and concrete examples makes the tasks descriptions accessible to those unfamiliar with the formal notation.

As mentioned in the previous section, the main drawbacks of taking a formal approach to task specification surround the lack of involvement of people. In particular, this means that additional work is required to assess and validate:

(1) whether the model is sufficient with respect to task coverage
(2) whether the tasks are 'real world' or constructs of the formal process

These aspects of the task classification developed in this thesis are addressed in Chapter 6. In addition, as noted in Section 3.1.2.1, typologies and design spaces do not provide information relating to which are the most frequently occurring and/or most useful tasks. Additional work is required to establish this.

3.3 **Requirement for extension to the Andrienko framework**

As mentioned briefly in Section 3.2, and discussed in more detail in Section 3.4, the Andrienko framework takes a formal approach to modelling the data and tasks involved in Exploratory Data Analysis. While the Andrienkos' interests lie in modelling spatial and temporal data, their framework is intended to be applicable to *all* types of data. One consequence of this is that the abstract nature of their resulting task categories proves too generic to use as a basis for mapping the visual techniques for their support. The initial intention in this work was simply to apply the Andrienko model directly to a particular class of data sets – temporal graph data – in order to elucidate the range of possible tasks. Such a data specific task listing – while not domain specific - would be specific enough to use as the basis of a mapping to visual techniques. However, in trying to apply the Andrienko framework to graph data, it was not clear how this could be done (see Section 3.5). Personal correspondence with Natalia Andrienko, one of the authors of the framework, confirmed that graph data was not considered when the framework was developed, and possible ways in which the data model could be extended were discussed. The extension to the data model and task framework which were required for use with graph data is the subject of Chapter 4. This extended model is then applied to produce a set of tasks appropriate to temporal graph data, which is outlined in Chapter 5. The rest of this chapter briefly sets out the original Andrienko framework, and gives further details relating to the limitations of the framework when applying it to the graph case.

3.4 **The Andrienko data model and task framework**

The Andrienko framework [5] consists of a data model and task framework. The framework was designed to be applicable to all types of data, and is rather complex, therefore the reader is referred to the original text for full details. The task framework (Section 3.4.3) uses a functional approach to specify the different types of tasks which may be involved in EDA, resulting in a "task typology". Under their model, there are two components to every task: the target (unknown information) to be obtained, and the constraints (known conditions) that information needs to fulfil. A task therefore involves finding a target given a set of constraints. The data model (Section 3.4.2) identifies the data items that can participate in tasks as a target or constraint.

The concepts of the Andrienko framework are illustrated with reference to an example author publication data set, which is first outlined.

3.4.1 Example author publication data

In order to help illustrate the abstract concepts of the Andrienko framework, a simple example academic author publication dataset is used. The data consists of a set of authors affiliated to an academic institution. In each year, the academic department to which the author belongs is recorded, along with details of their publications. The data is illustrated in Table 6.

Author	Year	Publications	Publication count	Department
A	2014	a, b, c, d, e	5	Computing
A	2013	f, g	2	Computing
A	2012	h	1	Computing
B	2014	a, b	2	Computing
B	2013	f, i	3	Biology
B	2012	j, k, l, m	4	Biology
…	…			

Table 6 Example co-authorship data

3.4.2 The Andrienko data model

The Andrienko data model identifies five data items that can participate in tasks as a target or constraint: individual characteristics, individual references, sets of references, behaviours, and relations. These data items, and how they are related to one another in the data model, are now discussed.

referrers *characteristics*

Author	Year	Publications	Publication count	Department
A	2014	a, b, c, d, e	5	Computing
A	2013	f, g	2	Computing
A	2012	h	1	Computing
B	2014	a, b	2	Computing
B	2013	f, i	3	Biology
B	2012	j, k, l, m	4	Biology
…	…			

A reference *A characteristic value*

Figure 6 Referrers and characteristics in the author publication data set

The Andrienko data model firstly divides data into two parts: referential and characteristic components (illustrated in Figure 6). The values of these components are known as references and characteristics respectively.

Referential components (a.k.a. referrers) describe the context or domain in which the data was obtained. These are independent components of the data, as they can (potentially) assume arbitrary values. There are three main types of referrer considered under the model: time, space, and population (a set of objects). A dataset may contain more than one referrer: in the example data set, the referrers are the authors (a population referrer) and time.

Characteristic components (attributes) represent the values obtained in this context. These components are dependent, as their values are determined by the choice of values of the referrers (e.g. the value of 'publication count' depends on which author and year we are considering). Characteristics may be of any data type: numeric, ordinal, categorical, sets, etc. In the co-authorship data example, they are department (categorical), publications (set) and publication count (numeric).

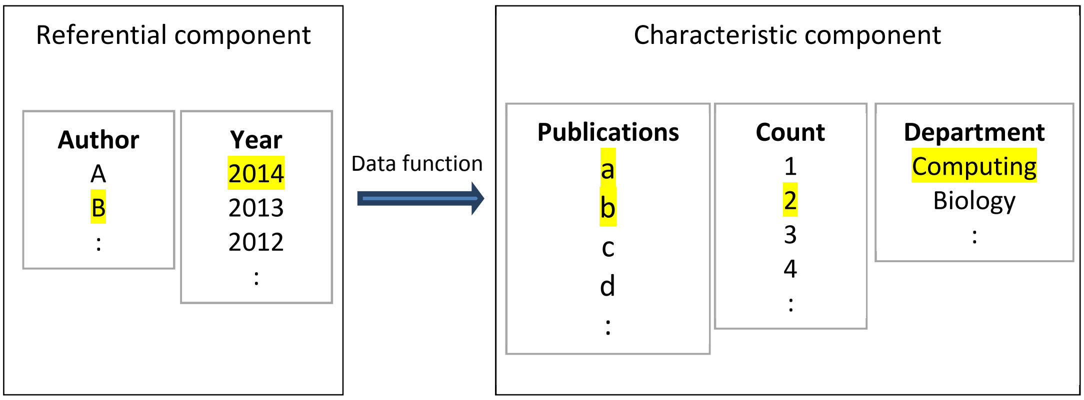


Figure 7 Illustration of the data function mapping between the referential and characteristic components; author B in 2014 (highlighted) has 2 publications, a and b, and belongs to the Computing department. The mapping can be described using formal notation e.g. $\boldsymbol{f}_{publications}$(2014, B) = {a,b}, $\boldsymbol{f}_{department}$(2014, B) = Computing **etc.**

The Andrienko framework considers the different ways in which data items can be related to one another: these are referred to as the **relations** between data components. Key to their data model is the correspondence between the referential and characteristic components. These components are related by the **data function** which is a mathematical metaphor to describe a simple look-up mapping between references and their corresponding characteristic value. This mapping can be written using formal notation, $\boldsymbol{f}(x) = y$, where x is an element of the referential component and y an element of the characteristic component. Figure 7 illustrates the data function using the author publication data.

In addition to the data function, relations also exist between data items within the referential and characteristic components. The **relations between individual characteristic values** depend on the underlying data type, and include equality, order, distance, and set relations. For example, we can ask whether two categorical data values are the same or different (equality); whether one ordinal value is greater than another (order); or of two numeric values, whether one is greater than the other (order) and if so, by how much (distance). Figure 8 gives examples of some possible relations between individual data components.

2014 is **2 years after** 2012 (distance, order)

2 publications are **2 less than** 4 publications (distance, order)

Authors are the **same** (equality)

Author	Year	Publications	Publication count	Department
A	2014	a, b, c, d, e	5	Computing
A	2013	f, g	2	Computing
A	2012	h	1	Computing
B	2014	a, b	2	Computing
B	2013	f, i	3	Biology
B	2012	j, k, l, m	4	Biology
…	…	…	…	…

Computing and Biology are **different** departments (equality)

These sets of publications **overlap** (set relations)

Figure 8 Examples of some possible relations between individual references and individual characteristics. Note that if we had two characteristic components which shared the same domain (for example, count of journal articles, and count of conference proceedings) we could also consider the relation between their individual values e.g. '3 journal papers is *1 paper greater than* 2 conference papers' (distance, order)

Relations between individual references also depend on their data type. Three types of relation are considered: continuity, order and distance. Time is continuous, ordered, with distances; space is continuous, unordered, with distances; population is discrete, unordered, without distance (see Table 7). How long (distance) a particular time point occurs before or after (order) another, or how far apart (distance) two locations are in space, is captured by the relations between references.

Subsets of the referential components are determined by the relations that exist between individual references. Because of the different types of relations that exist between their elements, each of the main referrer types has a different notion of what constitutes a **reference subset**. For example, time has time intervals: the elements between the start and end time instants are determined by the continuity and ordering relations. Other subsets include cycles in time, areas and lines in space, and groups of items in the population referrer. These reference subsets can also have relations between them. The **relations between reference subsets** include the set relations and those derivable from the relations between elements of the referrer. For example, the relations between time intervals could be described in terms of their temporal order, distance, and set relations (include, overlap, disjoint): the time period 1998-2004 is *two years before* the time period 2006-2008; 1998-2004 *overlaps with* the time period 2002-2006.[1]

		Referrer		
		Time	**Space**	**Population**
Elements		Time points	Locations	Any objects
Relations between elements	Order	Ordered	Unordered	Unordered
	Distance	With distance	With distance	Without distance
	Continuity[2]	Continuous	Continuous	Discrete
Subsets *(examples)*		*Time intervals*	*Areas, lines*	*Set of objects*
Relations between subsets		Order, distance, set	Distance, set	Set

Table 7 Summary of the relations within the three main referrer types

The final part of the Andrienko data model is that of **behaviours** (illustrated in Figure 9). So far we have noted that a reference subset (such as a time interval) is determined by the relations that exist between individual references. A reference subset also has a corresponding set of attribute values, as defined by the data function. Taken together, these relations – those that exist between references, and the mapping between references and characteristics (data function) - determine the configuration or arrangement of the corresponding characteristic values. For example, temporal ordering relations between time points determine the configuration of the set of characteristics over time: an author's publication count in

[1] Note that Lammarsch et al. [49] extend the Andrienko model with a detailed analysis of the structure of time and elucidate the full range of possibilities for relations within the temporal referrer.

[2] Note that Andrienko and Andrienko do not explicitly treat continuity as a relation in their model

2013, comes *before* that of 2014. This real-world phenomena is termed a behaviour in the Andrienko framework.

Patterns are subjective constructs resulting from an observation of a behaviour. They describe the "*essential features of a behaviour... in a substantially shorter and simpler way than specifying every(thing)"* ([5] p.85). For example, we might describe the behaviour of an author's publication count over time as an increasing or decreasing trend. A number of properties of patterns are outlined in the framework ([5] p90) including the degree of simplification; level of precision; coverage of the reference set (complete or partial); and the presence or absence of an overlap between sub-patterns. Four main types of pattern are also distinguished: association, differentiation, arrangement and distribution summary, although this is not intended to be an exhaustive list. Patterns are discussed further in Section 5.2.1.

Lastly, **relations also exist between behaviours**, and by extension, between patterns. For example, we could say that the trend in Author A's publication count (an increasing trend) is *opposite* to that of Author B (which is decreasing).

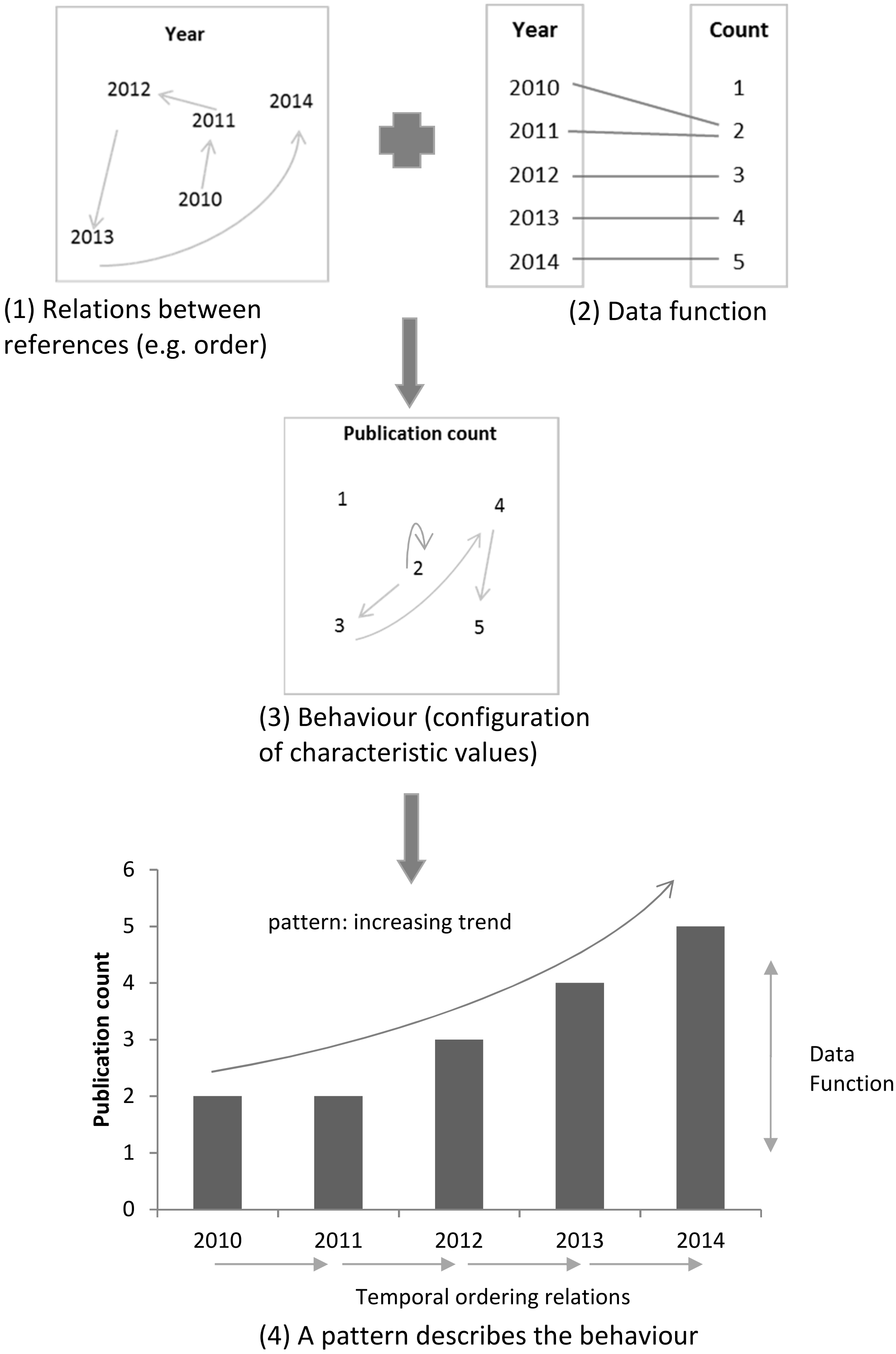


Figure 9 Behaviours and patterns: the behaviour (3), or configuration of characteristic values, is determined by the relations that exist between references in the referential component (1) and the

data function's mapping between individual references and corresponding characteristic values (2). A pattern such as an increasing trend (4), describes the behaviour.

3.4.2.1 Data model: summary

This section provides a brief summary of the components of the Andrienko data model, which are also illustrated in Figure 10.

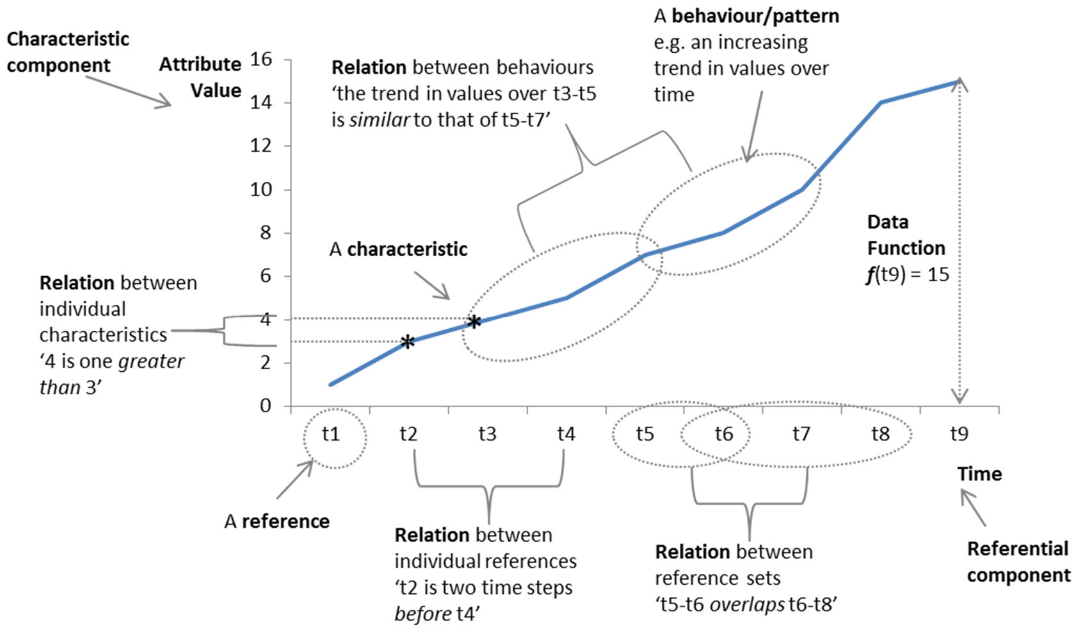


Figure 10 An illustration of the components of the Andrienko data model

Referential Component (referrers) – the independent data component (time, space, population)

- **Reference** – an individual item in the referential component e.g. a year, a point in space, a single element of a population.
- **Reference set** – a set of references e.g. a time interval, an area in space, a set of elements in a population

Characteristic component (attributes) – the dependent data component (may be of any data type: numeric, ordinal, categorical, sets etc.)

- **Characteristic** – an individual attribute value e.g. 10, first, red, x-small.
- **Behaviour** – the configuration of a set of characteristics which can be described by a **pattern** e.g. a temporal trend, a distribution in space, frequency of values in a population.

Relations - there are five possible relations (illustrated in Figure 11):

R1 Between references and characteristics (the **data function**)

Within the referential component:

R2.1 Between individual references (order, distance, continuity - see Table 7)

R2.2 Between reference sets (as for R2.1, plus set relations)

Within the characteristic component:

R3.1 Between individual characteristics (data dependent, including equality, order, distance, set relations)

R3.2 Between behaviours (similarity, difference, opposition, correlation, dependency and structural connection)

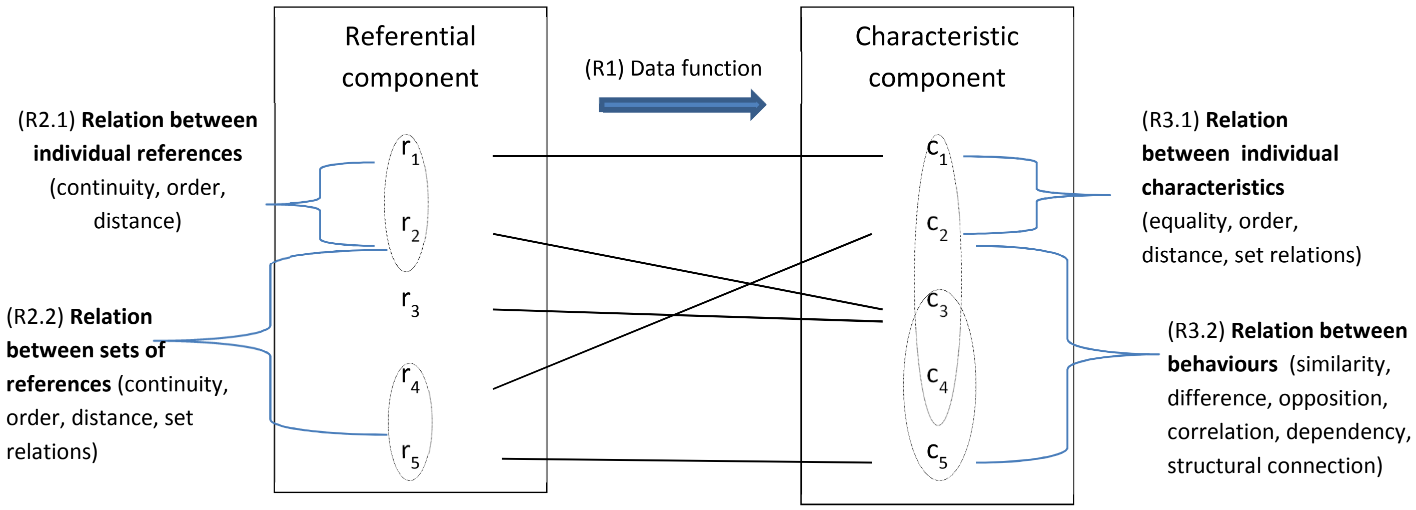


Figure 11 illustration of relations in the Andrienko data model. Data is divided into referential and characteristic components. The data function (R1) maps references to characteristic values; data dependent relations exist between references (R2.1), and subsets (R2.2) of the referential component; relations also exist between individual characteristics (R3.1), and behaviours (R3.2), in the characteristic component.

3.4.3 *The Andrienko task framework*

As mentioned at the beginning of this chapter, the Andrienko framework takes a functional approach to task specification. Under the framework, there are two components to every task: the target, or unknown information to be obtained, and the constraints, or known conditions, that information needs to fulfil. The five types of data item that can participate in a task as either a target or a constraint were distinguished in the previous section: individual characteristics, individual references, sets of references, behaviours, and relations.

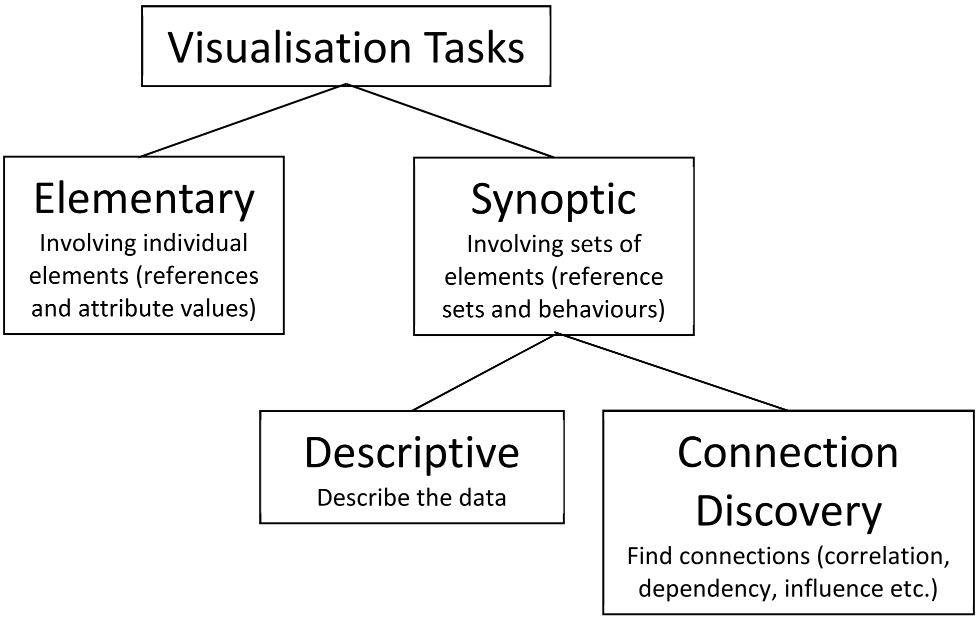


Figure 12 Visualisation tasks distinguished according to level of analysis

The tasks in the framework are distinguished according to the data items that participate in them. Firstly, in terms of the level of analysis (Figure 12): whether the task involves individual references and characteristics (*elementary tasks*), or sets of items (reference sets and behaviours) considered together as a unified whole (*synoptic tasks*). Synoptic tasks are further divided into descriptive tasks (concerned with describing or summarising the data) and connection discovery tasks (concerned with finding connections between phenomena, including correlation, dependency or influence, and structural connection). Secondly, tasks are distinguished according to which data items (referential components, characteristic components, or relations) participate as the task targets and constraints. This gives rise to three distinct task types: lookup, comparison, and relation seeking. In lookup tasks, the data function mapping is used to find the characteristic or referential component corresponding to a given data item. In comparison, the relation between two data items is the target. Relation seeking is the opposite of comparison, where the data items are the target, and the relation is a given constraint. The differences in targets and constraints between the task types are summarised in Figure 13, and each task type is described briefly, below. Note that each of these task types can take the form of an elementary or synoptic tasks. Tasks in the framework are specified using a formal notation. A description of this formal notation for each task can be found in Appendix A.

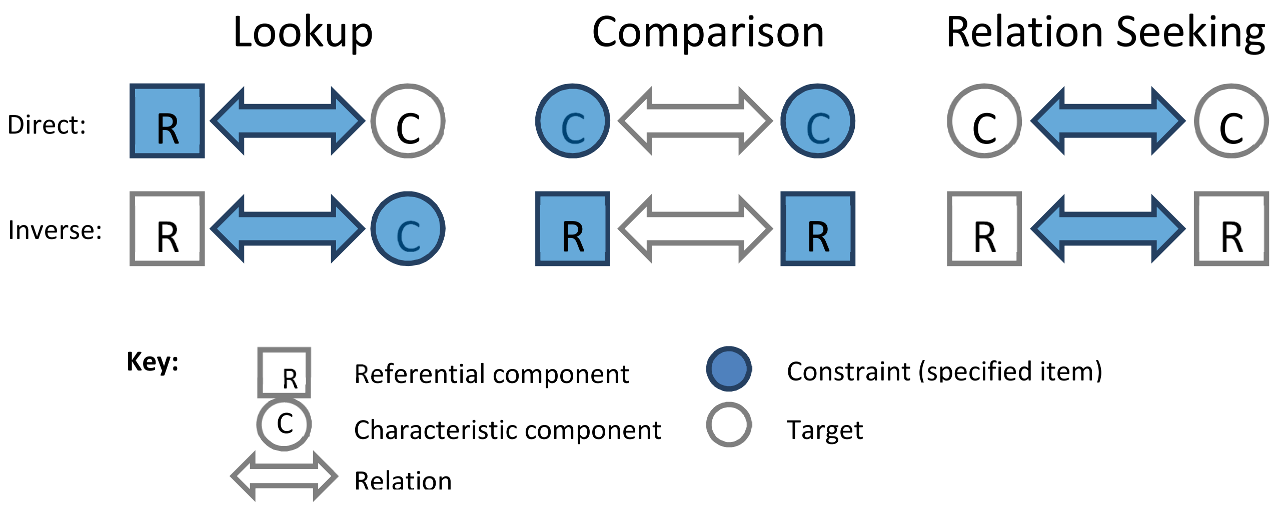


Figure 13 Three general task types are distinguished according to which data items participate as targets or constraints (indicated in white and blue in the figure, respectively). In lookup tasks, the data function, and a characteristic or referential component is specified: the task target is the corresponding referential or characteristic component. In comparison, the relation between two data items (characteristic or referential components) is the target. Relation seeking is the opposite of comparison, in this case the relation is known, and the task is to find data items which are related in the given way. Direct and inverse variations of the tasks are distinguished according to the referential and characteristic components involved.

3.4.3.1 Lookup

On elements, lookup involves finding a characteristic given a reference (direct lookup) or references given a characteristic value (inverse lookup). On sets it involves finding the pattern associated with the behaviour of an attribute over a reference set (behaviour characterisation), and inversely, finding the subset of references corresponding to a given pattern (pattern search).

Examples:

The following examples use the publication counts of an individual author over time to illustrate the lookup tasks. Items highlighted in yellow in the illustrations are known items (constraints) while those surrounded by a dotted line are the targets.

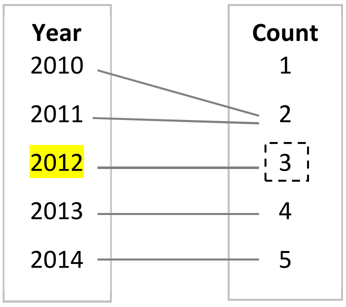


Figure 14 Elementary direct lookup

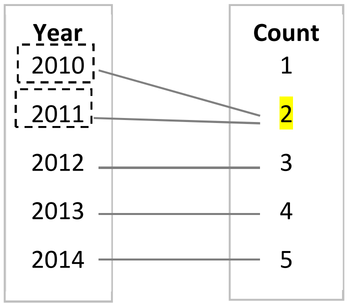


Figure 15 Elementary inverse lookup

Direct lookup, elementary (Figure 14): *How many publications did the author have in 2012*?

Inverse lookup, elementary (Figure 15): *In what year(s) did the author publish two publications?*

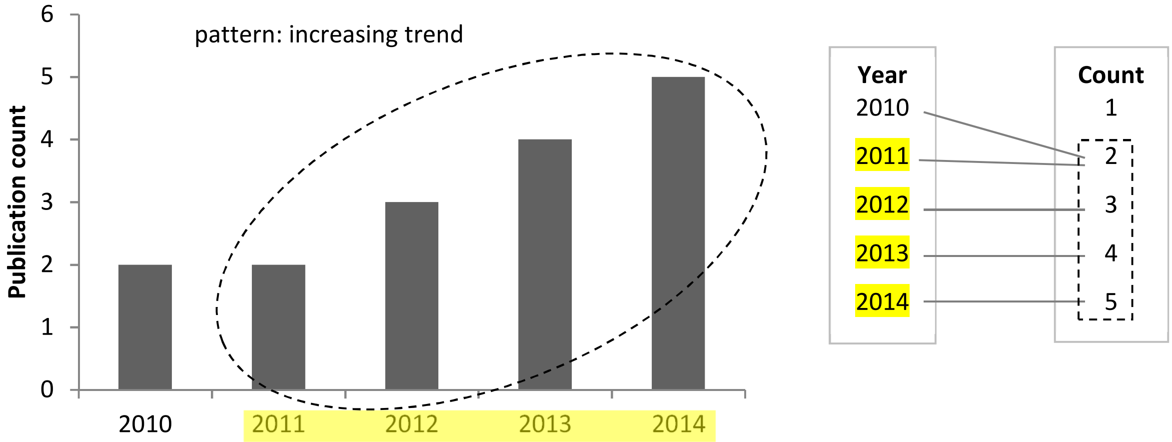


Figure 16 Behaviour characterisation

Behaviour characterisation (direct lookup on sets) (Figure 16): *What was the pattern in the author's publication count between 2011 and 2014?*

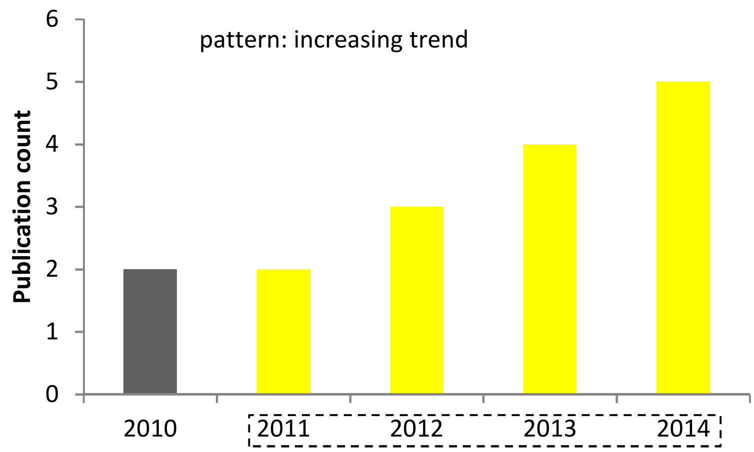


Figure 17 Pattern search

Pattern search (Figure 17): *Find the time interval over which there was an increasing trend in the author's publication counts.*

3.4.3.2 Comparison

Comparison involves finding the relation between specified components; either between characteristics or patterns (direct comparison), or references or reference sets (inverse comparison). Under the Andrienko framework, comparison is a compound task, as it always requires at least one lookup task to find one of the data items being compared. This is because comparing known values in isolation, for example, red and blue, or the years 1980 and 1981, is not an analytical task; the answer will always be the same and is known without having to investigate the data. A useful comparison is one where at least one of the values involved is dependent on the data function mapping, for example, comparing the author's publication counts in 2013 with those of 2014. We will return to this point in Section 4.2.

A number of variations of comparison tasks are outlined in the Andrienko framework, based on the constraints involved. These are listed in Table 8 for reference.

	Elementary	Synoptic
Direct comparison…	With specified attribute values	With a specified pattern
	Between values of the same attribute(s) for different references	Between behaviours of the same attribute(s) over different reference sets

	Between values of different attributes for the same reference	Between behaviours of different attributes over the same reference set
	Between values of different attributes for (partly) different references	Between behaviours of different attributes over (partly) different reference sets
Inverse comparison…	With specified reference(s)	With specified reference sets
	Between references corresponding to different values of the same attribute(s)	Between the reference sets corresponding to specified behaviours of the same attribute(s)
	Between references corresponding to specific values of different attributes	Between the reference sets corresponding to specified behaviours of different attributes

Table 8 Variations of comparison tasks, extracted from [5] p121-3.

Examples:

The following examples use the publication counts for two authors, author A and author B, over time. Again, items highlighted in yellow in the illustrations are known items (constraints) while those surrounded by a dotted line are the targets.

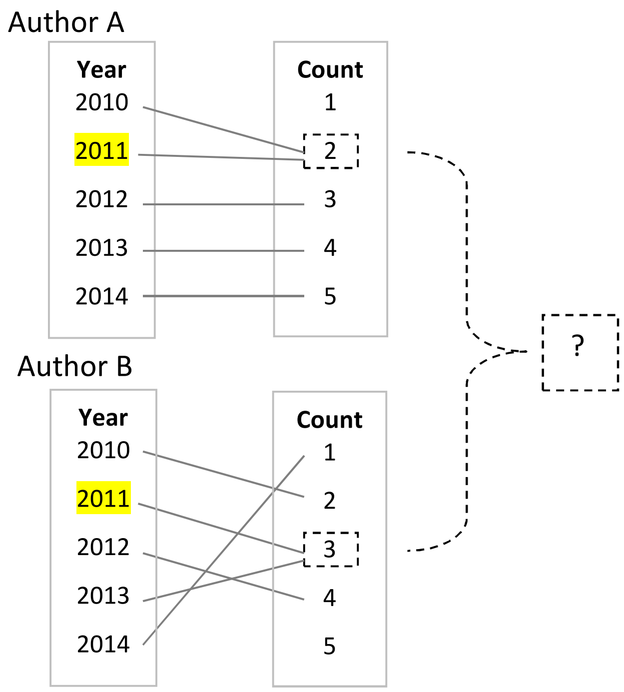


Figure 18 Elementary direct comparison

Elementary direct comparison (Figure 18): *compare author A and B's publication counts in 2011.*

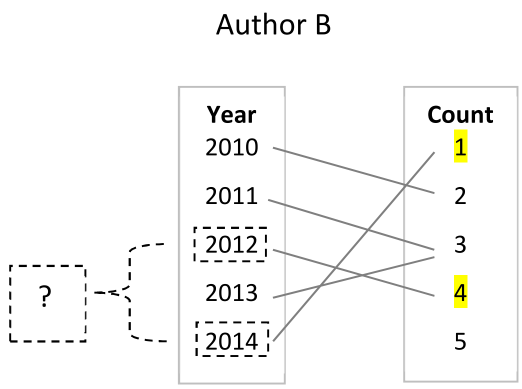


Figure 19 Elementary inverse comparison

Elementary inverse comparison: *compare the years in which author B had their highest number of publications (4) and their lowest number of publications (1).*

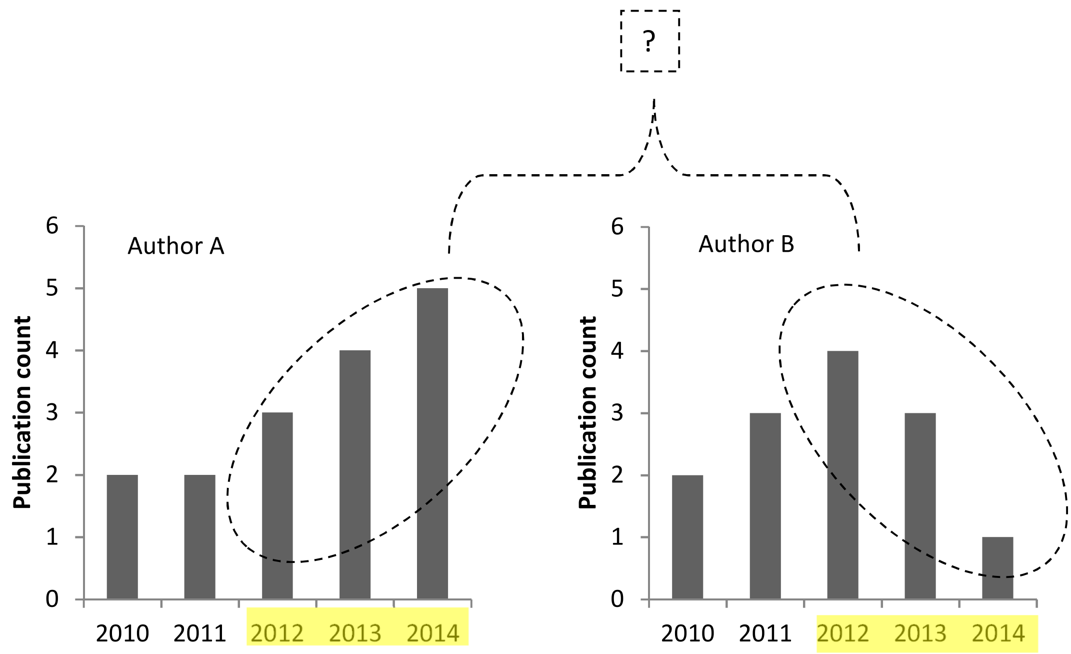


Figure 20 Synoptic direct comparison

Synoptic direct comparison (Figure 20): *compare the trend in author A and B's publications between 2012 and 2014.*

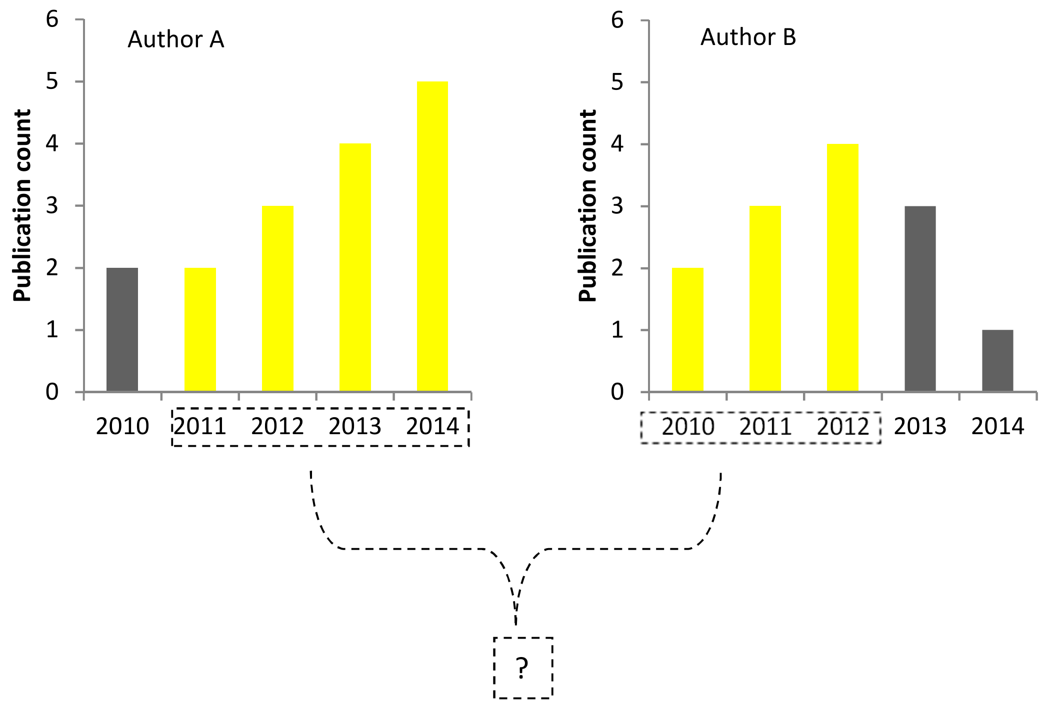


Figure 21 Synoptic inverse comparison

Synoptic inverse comparison (Figure 21): *compare the time periods over which authors A and B had increasing trends in their publication counts*.

3.4.3.3 Relation seeking

Relation seeking is essentially the opposite of comparison, where we wish to find components associated by a specified relation. Like comparison, it is a compound task which requires at least one lookup task.

Andrienko and Andrienko note that to specify a relation alone is unusual in practice, and an additional constraint is typically required. They therefore offer four additional variations of this task, included in Table 9 for reference.

Elementary relation seeking…	Synoptic relation seeking…
Between values of attribute(s) and, at the same time, between references	Between behaviours of attribute(s) and, at the same time, between reference sets
Between characteristic(s) of a specified reference and characteristics of other references	Between an attribute behaviour over a specified reference subset and attribute behaviours over other reference subsets
Between values of the same attribute(s) for partly different references (in a dataset with multiple referrers)	Between behaviours of the same attribute(s) over partly different

	reference sets (in a dataset with multiple referrers)
Between values of different attributes for the same reference	Between behaviours of different attributes over the same reference set

Table 9 Variations in relation seeking tasks, extracted from [5] p123-4

Examples:

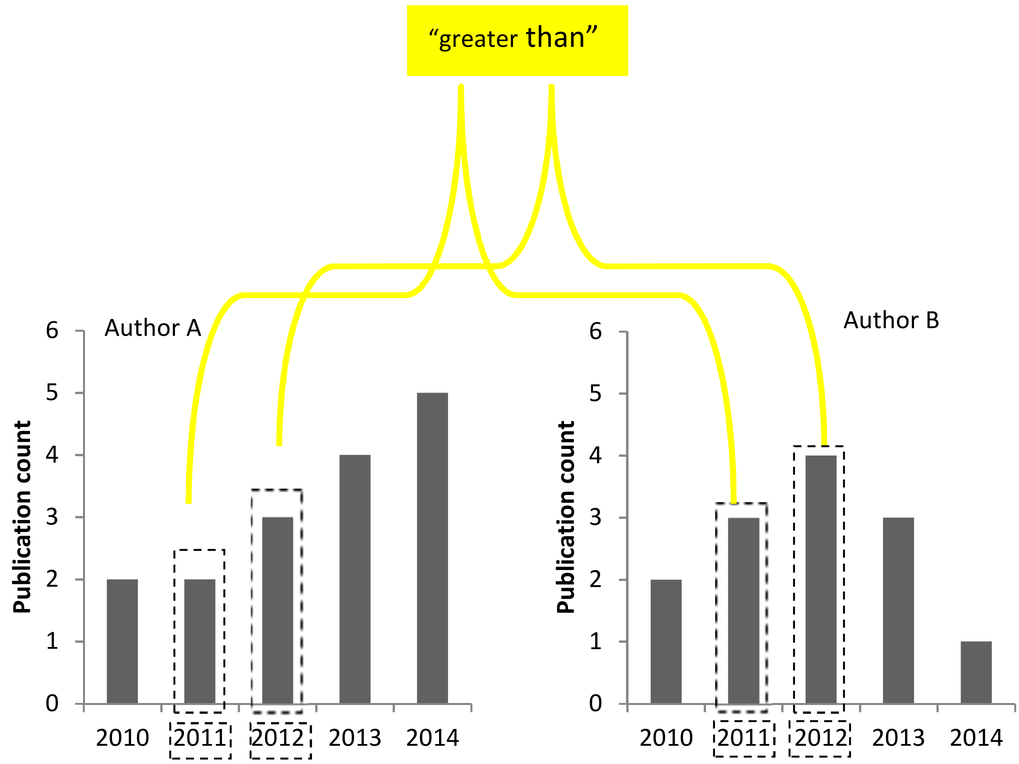


Figure 22 Elementary relation seeking: a relation between elements is specified. In this case the relation specified is between characteristic values. The target(s) are the corresponding references which are found using inverse lookup tasks.

Elementary relation seeking (Figure 22): *find the times at which author B's publication count was greater than author A's.*

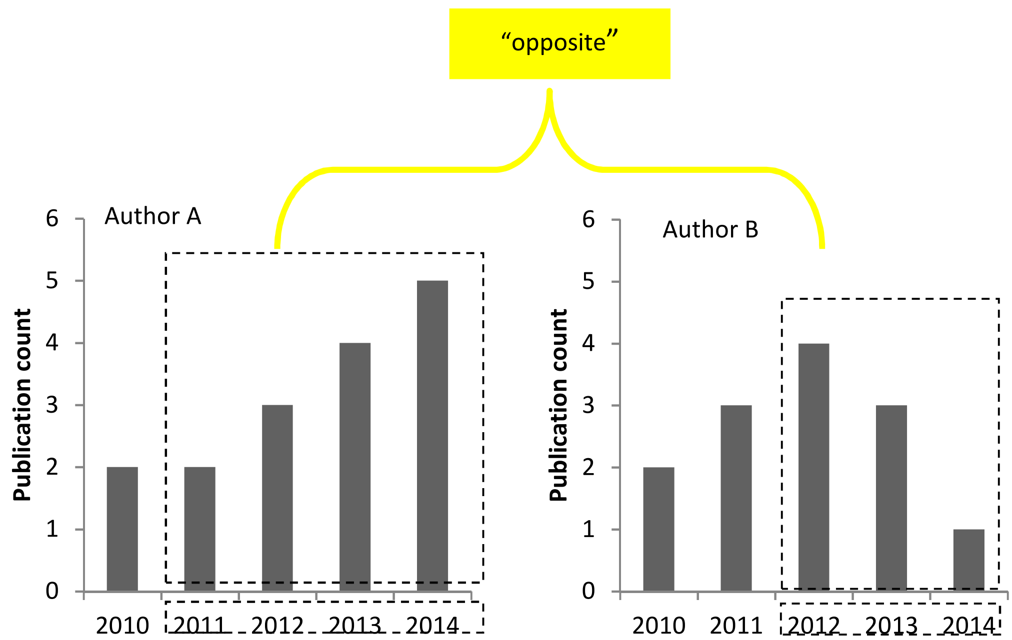


Figure 23 Synoptic relation seeking

Synoptic relation seeking (Figure 23): *Find time periods during which the authors had opposite trends in publication counts*

3.4.3.4 Connection Discovery

All of the examples given above are of descriptive tasks, as they simply describe the data. One final, but most important set of tasks considered in the Andrienko framework are the connection discovery tasks. These tasks also involve behaviours, but they do more than just describe the occurrence of phenomena (as is the case with behaviour characterisation in the descriptive tasks). Their aim is to find indications of possible connections or relations either between the parts of a single phenomenon (homogeneous behaviours) or between two or more phenomena (heterogeneous behaviours). In these tasks, we are interested in two or more behaviours with respect to each other. Such behaviours are termed 'mutual' or 'relational' behaviours, and can be described using one of three 'linkage patterns': correlation, dependency or influence, or structural connection (i.e. the interplay of two components, such as a trend over time and variation over seasons). Three variations of these relational behaviours are described in the framework based on the items between which the relations occur: (1) two (or more) different attributes of the same reference set; (2) two (or more) different attributes of different reference sets; and (3) the same attributes of different reference subsets. Tasks involving these behaviours can be formulated for each of the three main task types. The connection discovery tasks are discussed in more detail in Section 5.6.

3.4.3.5 Task framework: summary

Aigner et al. [98] show the tasks of the Andrienko framework organised into a taxonomy (redrawn in Figure 24).

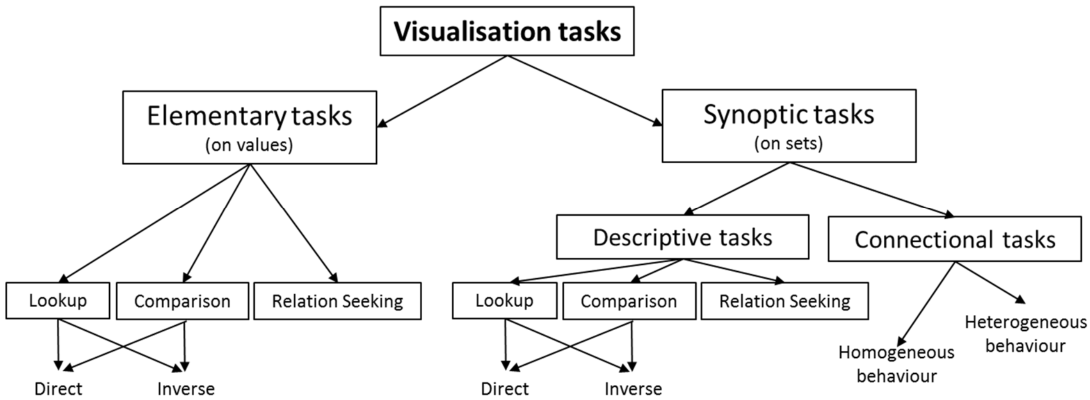


Figure 24 The Andrienko task model organised into a taxonomy. Redrawn from Aigner et al. ([98] **p74).**

Figure 25 illustrates the combined dimensions (level of analysis and task type) for the descriptive tasks of the taxonomy, with example tasks.

Figure 25 Illustrating the combined dimensions (level of analysis and task type) of the Andrienko task framework

**at least one of these components is found via a lookup task*

Task type		**Target**	**Constraint**	**Elementary (example)**	**Synoptic (example)**
Lookup	Direct	**characteristic**	**referential**	To which department did Author A belong in 2012?	What is the trend in Author A's publication counts 2012-2014?
	Inverse	**referential**	**characteristic**	Which author(s) had more than 4 publications in any year?	Find authors who move frequently between departments.
Comparison	Direct	**relation**	**characteristic***	Compare the publication counts of Authors A and B in 2014.	Compare the trend in Author A's publication counts for 2012-2014 with the trend for 2009-2012.
	Inverse	**relation**	**referential***	Did Author A's highest publishing count occur before or after his lowest?	Compare the time periods over which Author A's publication counts were increasing with the time periods over which they were decreasing.
Relation seeking		**characteristic**/ **referential***	**relation**	Find the year in which Author B moved departments (i.e. consecutive years where Author B belonged to two different departments).	Find authors with similar patterns in movement between departments.

3.5 Limitations of the Andrienko framework

Although the Andrienko framework is intended to be applicable to all types of data, the application of the framework to some data types requires further consideration. Recently, Lammarsch et al. [49] extended the framework to support task formulation for time-oriented data analysis, by developing a rule set that explicitly models the structure of time. As discussed in Section 3.2, the Andrienko framework does not consider graph data. In order to be usable with graph data, it was necessary to extend both the data model and task framework.

Let us first discuss why an extension to the data model is necessary. Modelling edges proves difficult under the existing framework, the problematic question being: what type of data item is an edge? An intuitive answer is that edges are relations between references (nodes). However, the types of referrers and relations considered under the data model are not sufficient to represent this. This can be demonstrated with reference to the author publications data example.

Author	Year	Publications	Publication count	Department
A	2014	**a**, **b**, c, d, e	5	Computing
A	2013	**f**, g	2	Computing
A	2012	h	1	Computing
B	2014	**a**, **b**	2	Computing
B	2013	**f**, i	3	Biology
B	2012	j, k, l, m	4	Biology
…	…	…	…	…

Figure 26 A co-authorship network can be extracted from the author publication data set based on authors who have publications in common

In our author publications data set, we may wish to extract and consider a co-authorship network (Figure 26). Using the data model, we can consider authors (nodes) in the network to be references; the open question is how to represent the co-authoring edges. The task, *did Authors A and B co-author in 2012?,* strongly resembles a comparison task i.e. *find the relation between author A and B in 2012*. This would suggest that edges be modelled as relations between references. However, the relations between references of the main referrer types considered under the data model (see Table 7) are insufficient to describe edges. As the authors of the network are clearly neither temporal nor spatial in nature, population is the remaining option for referrer type. The elements of a population referrer are discrete, unordered, and without distance. While these relations are appropriate when considering an unconnected set of objects, they are not sufficient to capture the co-authoring relations (edges) which exist between authors.

To model edges, it is therefore necessary to extend the Andrienko data model. A new referrer type (graph) is introduced, along with a new type of relation (linking) which exists between its elements. As a result of the extension to the data model, a set of structural tasks for use with the graph referrer type are also posited, thereby extending the task framework. The extensions to the framework are presented in Chapter 4.

3.6 **Summary**

This chapter has reviewed the possible approaches to developing a task classification and associated threats to validity at each stage (Section 3.1). It has set out the reasons for the approach to task classification construction adopted in this thesis, considering both the advantages and limitations of the chosen strategy (Section 3.2). The Andrienko framework, upon which the task taxonomy in this thesis is based, was outlined (Section 3.4), and the limitations with regard to modelling graph data under this framework were discussed (Section 3.5). The extensions to the framework which address these limitations are the subject of the next chapter.

Chapter 4 Extension of the Andrienko Framework for Graph Data

In Chapter 3, the limitations of the Andrienko framework [5] with regard to modelling graph data were outlined. This chapter presents an extension to the framework for use with graph data: Section 4.1 details the extensions to the data model, and Section 4.2 outlines the extensions to the task framework. The chapter concludes with a complete listing of the tasks for graph data under the extended task framework.

4.1 Extensions to the data model[3]

Two extensions are made to the data model: the introduction of a new referrer type, 'graph', and a new type of relation, 'linking', which exist between elements of the graph referrer. The example of a co-authorship network which can be extracted from publication data, as outlined in Chapter 3, is here continued to help illustrate these ideas.

4.1.1 Linking relations

'Linking' relations exist between elements of a graph referrer. These relations are specified by the edges between nodes. They are asymmetric (in an unordered graph, one edge (a,b) specifies two linking relations i.e. from a to b and from b to a), and can be viewed as qualitative (exists or not) or quantitative (expressed numerically in terms of the strength of the link (link weight), where 0 means no link). Further – and unlike the other relations between references - they may change over time in terms of their existence or strength. Linking relations may also have domain properties associated with them, such as an edge type.

In the example author publication data set outlined in Chapter 3, a linking relation represents the co-authoring relation between two authors in the extracted co-authorship network (illustrated in Figure 27).

[3] Note that some of the ideas relating to extending the data model are based on discussion with Natalia Andrienko via personal communication [263].

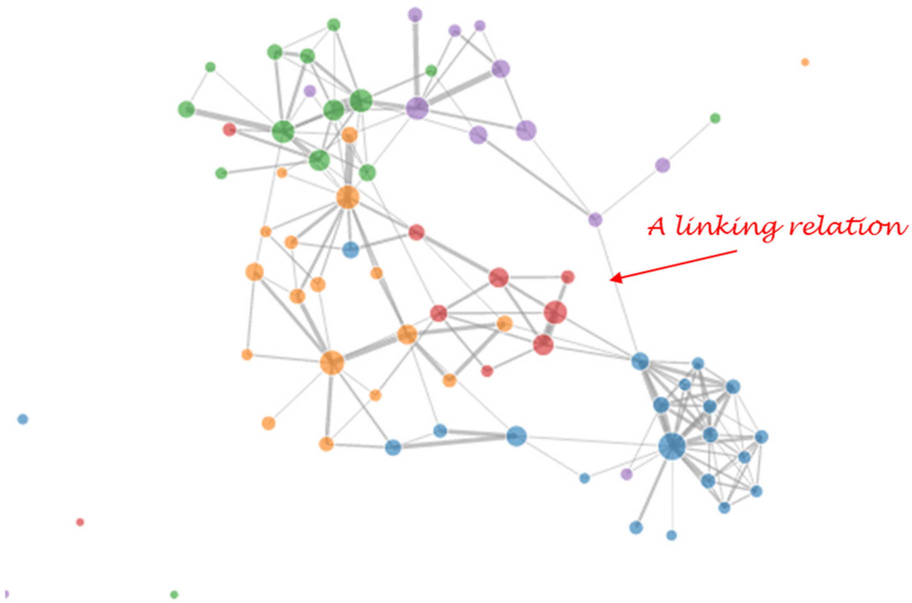


Figure 27 An example co-authorship network represented as a node-link diagram. Nodes (circles) represent authors, edges (lines connecting nodes and their thickness) represent the level of co-authorship between two authors. Size of node encodes and author's publication count; colour indicates the department to which they belong. In the extended data model, linking relations are specified by the edges between nodes.

4.1.2 Graph referrer

The graph referrer is distinguished from space, time, and population referrers, by the type of relations which exist between its elements: graph is discrete, unordered, with distances, and has linking relations. Table 10 shows a summary of referrer types extended to include the graph referrer. The distance relation between elements in a graph is dependent upon the linking relations. Distance between two elements can be defined as the geodesic distance (i.e. the number of edges in the shortest path between two nodes).

		Referrer			
		Time	**Space**	**Population**	**Graph**
Elements		Time points	Locations	Any objects	Nodes
Relations between elements	Order	Ordered	Unordered	Unordered	Unordered*
	Distance	With distance	With distance	Without distance	With distance
	Continuity	Continuous	Continuous	Discrete	Discrete
	Linking	Without links	Without links	Without links	With links
Subsets *(examples)*		*Time intervals*	*Areas, lines*	*Set of objects*	*Graph objects e.g. cluster, path*
Relations between subsets		Order, distance, set	Distance, set	Set	Distance, linking, set

**ordering is present when dealing with paths in a directed graph*

Table 10 Summary of the properties of referrer types, extended to include the graph referrer and linking relations (highlighted in yellow)

In addition to direct connections, indirect connections, or transitive relations, exist between elements of the graph referrer. These can be described in terms of the "chain" of linking relations. A transitive linking relation has the same properties as (direct) linking relations: existence, a direction, and possibly a weight/domain specific property (or some aggregated notion of weight, based on the weights of the individual connection relations), plus a distance between elements. It may additionally take into account edge weights. The meaningfulness of distance in transitive relations is domain dependent.

In the example author publication data set, when we extract the co-authorship network we are treating the set of authors as a graph referrer.

As discussed in Section 3.4.2, subsets of a reference set can be defined based on the relations that exist between elements: a subset of time is a time interval; the elements belonging to the time interval are determined by the temporal referrer's ordering relations. In the same way, subsets of the graph referrer are defined based on the linking and distance relations that exist between its elements. These subsets can be referred to as 'graph objects': a subset of nodes, which have a set of linking relations (edges) between them. Examples of graph objects include Lee et al.'s [40] graph specific objects: paths, groups, connected components, and subgraphs. Note that while the nodes of the graph referrer are unordered, an additional ordering relation is present when dealing with paths and directed graphs. Due to the nature of the linking relations between elements of the graph referrer, these subsets are not fixed (as is the case for the other referrer types).

In the co-authorship network example, a group of authors who publish together would be an example of a graph object (Figure 28).

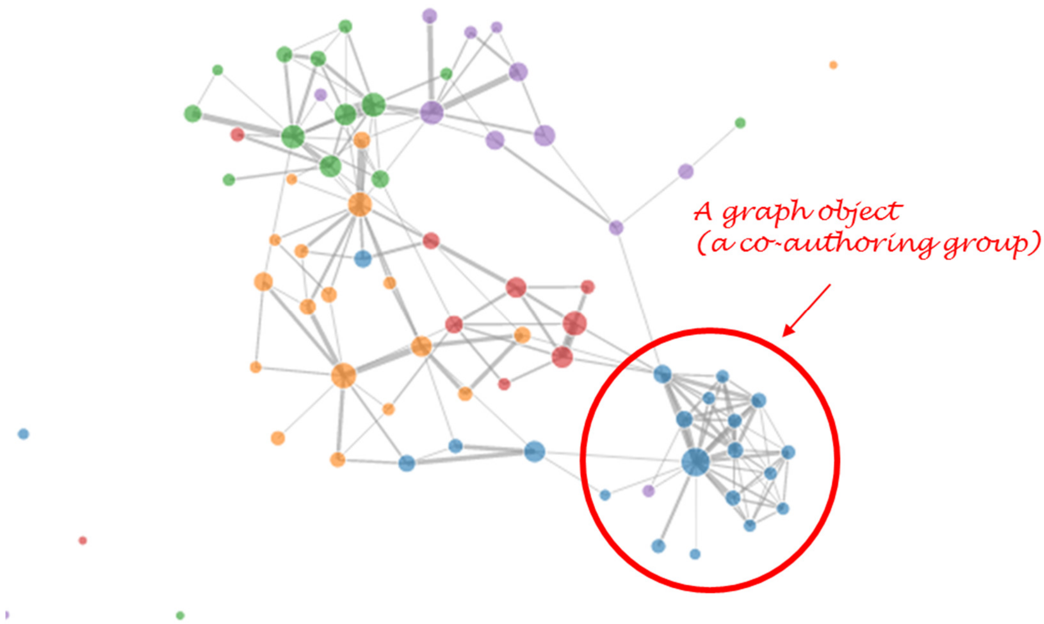
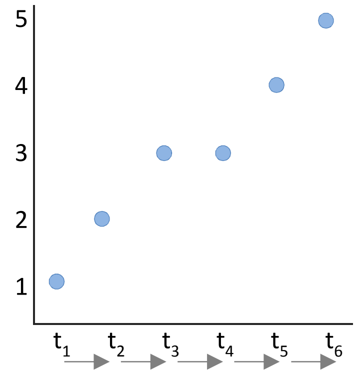
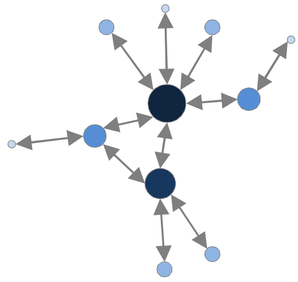


Figure 28 An example of a graph object - a co-authoring group - in the co-authorship network

The relations between graph objects (or a graph object and a node) are linking relations, distance, and the set relations (include, overlap, disjoint). For example, two clusters may be connected directly or indirectly (linking relation, distance); their elements may overlap, be entirely disjoint, or one cluster may include other sub clusters (set relations).



Ordering relations between time points determine the order in which both time points and their corresponding attribute values appear in the time series, producing an increasing trend over time.



Linking relations between nodes determine both the structure of the graph and the position in which attribute values appear.

Figure 29 Illustrating the role of relations in the time and graph referrers.

Behaviours (configurations of attribute values) are in part determined by the relations which exist between references. In the same way that temporal trends in attribute values are determined by the ordering relations between time points, in the graph case, the distribution of attribute values over the graph structure is determined by the linking relations between the nodes of the graph referrer (illustrated in Figure 29). Patterns describing the behaviours of the department and publication count attributes over the co-authorship network shown in Figure 27 might include that ‘more central authors have higher publication counts’ and ‘authors belonging to the same department tend to co-author together’.

As linking relations between references are not fixed, one final extension to the data model is made, that of structural behaviours and structural patterns. **Structural behaviours** are closely related to the original Andrienko notion of behaviour: they are the configurations of references (nodes), as determined by the linking relations between them. For example, authors belong to a co-authoring cluster by virtue of the co-authoring (linking) relationships that exist between them. **Structural patterns** describe structural behaviours and include clusters, cliques, motifs and network structures (small world, scale-free etc). In our co-authorship network we might describe a group of authors who all co-author with one another as a tightly connected co-authoring cluster, or describe a group of authors who are connected by virtue co-authoring with a single central authors as forming a star motif (Figure 30).

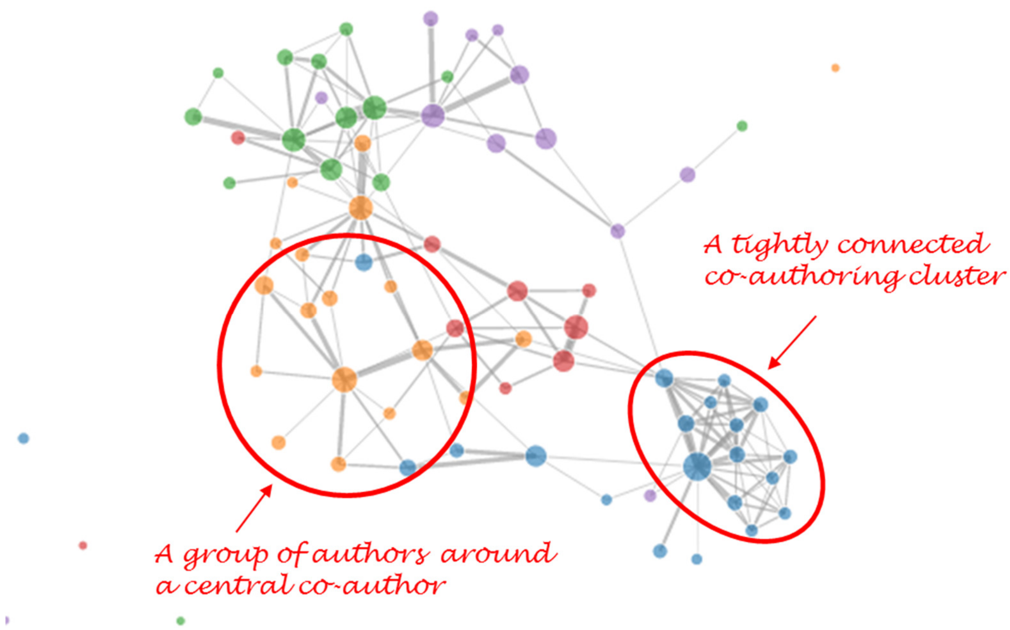


Figure 30 Structural behaviours and patterns in a co-authorship network. Structural behaviours, such as a co-authoring cluster, are determined by the co-authoring (linking) relations that exist between authors. These are described by structural patterns, such as a star motif (left) where a group of authors all co-author with a central author, or a tightly connected co-authoring cluster (right).

A summary of how data model terms apply to graphs is given in Table 11.

Data model term*	**Graph term**	**Co-authorship network example**
A reference	node	An author
(*linking*) relation	edge	The co-publishing relationship between two authors
A characteristic	an attribute value	An author's publication count, the research centre to which they belong
A reference set	a set of nodes	A set of authors
Structural behaviour	graph objects e.g. path, cluster, subgraph etc.	A group of co-authors
Structural pattern	a cluster, clique, small world network etc.	A pattern to describe the structure of a co-authoring group e.g. a tightly connected co-authoring cluster or a set of authors

		grouped around a central author (star motif)
A behaviour (described by a pattern)	distribution of attribute values over the graph	'More central authors have higher publication counts'; 'authors belonging to the same department tend to co-author together'

* additions to the Andrienko data model are shown in italics

Table 11 Relating data model terms to graph terms

4.2 Extension to the task framework

As outlined in Section 4.1.1, under the extended data model, edges are modelled as relations between references. Relations between references appear in the inverse comparison and relation seeking tasks of the Andrienko framework. This means that in the graph case, we can formulate questions such as *are the authors with the highest and lowest publication counts co-authors*? (inverse comparison) and *which of author A's co-authors belong to a different department?* (relation seeking). Treating edges as relations also allows us to apply the Andrienko notions of behaviour and pattern to the graph case (as illustrated in Figure 29). Thus we can find, describe, and compare these attribute patterns and behaviours, and their associated subgraphs, using the synoptic tasks of the existing framework.

What is important to note, however, is that all of the tasks in the Andrienko framework involve the data function, that is, they always require at least one lookup task involving an attribute value. Yet in the graph case, there are simpler tasks which involve only the graph's structure, for example, *are authors A and B co-authors?* and *who are author A's co-authors?*. Modelling these tasks - which involve only the relations between references - requires an extension to the task framework. This is outlined in Section 4.2.2. In addition to investigating the relations between two graph objects, we may also be interested in the relations within a set of nodes: how the nodes are connected, and whether a particular pattern, or configuration of connection is apparent. A further extension is therefore made to the task framework in order to accommodate tasks involving the structural behaviours and structural patterns introduced under the extended data model. This is outlined in Section 4.2.3.

4.2.1 'Pure' relational tasks in the Andrienko framework

When outlining the tasks in their framework, Andrienko and Andrienko consider a set of "pure relational tasks". These tasks involve only the relations between elements, and may be constructed according to one of the following general schemes :

1. How are the elements **p** and **q** (or the subsets **P** and **Q**) of the set **S** related?
2. What element (or subset) of the set **S** is related to the element **p** (or subset **P**) in the way **ρ**?
3. What elements (or subsets) of the set **S** are related in the way **ρ**?

(Andrienko & Andrienko [5] pp. 62-63)

[Note that pure relational task (1) is the comparison subtask ?λ: **p** λ **q**, while (2) and (3) are the two possible variations of the relation seeking subtask ?*q*: **pΛ***q* and ?*p, q*: *p***Λ***q*, respectively.]

These questions, we are told, "*address general properties of the sets from which the references and characteristics are taken and have no relevance to any particular dataset*" ([5] p. 63). For example, answering '*how are years 1980 and 1981 related?*' does not provide us with any new insight: 1980 is always the year prior to 1981 and we do not require a dataset to know this. As such, they are not typical of data analysis and are therefore not included in the framework as stand-alone tasks; hence the requirement for at least one look-up task in the comparison and relation seeking tasks.

However, the Andrienko framework did not consider graph data when it was developed. In the extended data model, linking relations[4] between elements of the graph referrer are not fixed: they differ depending on the data set, and even within a dataset, may change over time. This introduces a level of unpredictability into the referential component of the data set, and as such, without investigating our data we

[4] and resultantly, distance relations and set relations.

cannot answer these relational questions. It therefore makes sense to formulate these questions in the graph case, for example:

- *How are nodes **p** and **q** (or subgraphs **P** and **Q**) of the graph related?*
- *Which node(s) of the graph are connected to node **p** at a **distance of less than or equal to n**? To which cluster does node **p** belong?*
- *Which nodes of the graph are **directly connected**? Which clusters of the graph **overlap**?*

The task framework is therefore extended to account for this feature of graph data.

4.2.2 Extension: structural comparison and relation seeking tasks

The "pure" comparison and relation seeking tasks are included in the extended task framework. To help differentiate them from the inverse comparison and relation seeking tasks of the original framework (which involve the data function) they are referred to as *structural comparison* (scheme 1, outlined in Section 4.2.1) and *structural relation seeking* (schemes 2 and 3). Structural comparison involves finding relations between graph objects, while structural relation seeking concerns finding graph objects related in a given way.

Three variations of each of the Andrienko schemes can be considered according to the combination of nodes and/or graph objects involved:

- Tasks involving two nodes
- Tasks involving a node and a graph object
- Tasks involving two graph objects.

The relations that can be considered in each task depend on these combinations:

- Between two nodes: linking, distance, and order (in directed graphs)
- Between a node and a graph object, or between two graph objects: linking, distance, order (in directed graphs), and set relations.

Figure 31 illustrates structural comparison and relation seeking involving nodes.

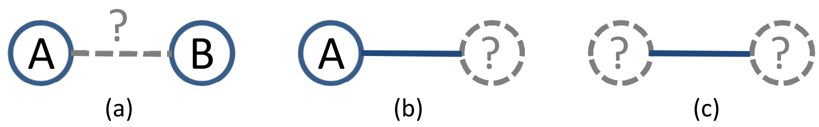


Figure 31 structural comparison and relation seeking involving nodes. (a) structural comparison (scheme 1): find the relation between given nodes. (b) structural relation seeking (scheme 2): find nodes related to the given node in the given way. (c) structural relation seeking (scheme 3): find nodes related in the given way

Linking relations can be specified in a number of ways: qualitatively (in terms of their existence), quantitatively (in terms of link strength or weight), with reference to direction (in directed graphs), and possibly domain properties (such as edge type). The possible set relations are include, overlap, and disjoint[5].

Based on the combinations of scheme (comparison or relation seeking), whether nodes and/or graph objects participate, and the relations of interest, a wide variety of tasks can be constructed. Some examples of variations in structural relation seeking (scheme 2) are suggested in Figure 32.

Linking relations:
What node(s) are connected to node **p**? *Which author(s) co-author with author A?*
What clusters(s) are connected to node **p**? *With which co-authoring group(s) has author A co-published?*
What clusters (s) are connected to cluster **P**? *Which co-authoring group(s) are connected to group X?*
What node(s) are connected to node **p**, with a weight greater than 2? *Which author(s) have co-authored with author A at least twice?*
What node(s) have a connection *from* node **p**?
What node(s) have a friendship relation with node **p**? *Which author(s) have co-authored a book with author A?*

Distance relations:
What node(s) are connected to node **p** at a distance of less than or equal to n? *Who are author A's co-authors' co-authors?*

Set relations:
Which graph objects (nodes, clusters, subgraphs, paths etc.) belong to subgraph **P**? *Which authors belong to co-authoring group X?*

[5] Between a node and a graph object, there is no notion of overlap: a node either belongs to the graph object or does not.

Which graph objects (clusters, subgraphs, paths etc.) overlap with subgraph **P**? *Which authors belong to co-authoring group X and also co-authoring group Y?*
Which graph objects (nodes, clusters, subgraphs, paths etc.) are disjoint with subgraph **P**? *Which authors are not connected to the main co-authorship network?*

Figure 32 illustrating some possible variations of Andrienko scheme 2, *What element (or subset) of the set S is related to the element p (or subset P) in the way ρ?* (relation seeking), according to the relations and graph objects involved. Note that combinations of linking/distance/ordering relations are also possible e.g. *What node(s) have a friendship relation of strength 4 from node p?*

4.2.3 Extension: tasks involving structural behaviours and structural patterns

Under the extended data model, structural behaviours and structural patterns were introduced in order to capture the configurations of connectivity that are possible between elements of the graph referrer (Section 4.1.2). A set of tasks are therefore required in order to describe and explore these structural behaviours and patterns. These tasks are almost identical to the synoptic tasks of the existing task framework (outlined in Section 3.4.3), but they involve structural patterns and behaviours (detailed variations and examples are listed in Section 4.3). Note that the figures used to illustrate the tasks follow the same format as those of Section 3.4.3 (items highlighted in yellow in the illustrations are known items (constraints) while those surrounded by a dotted line are the targets.).

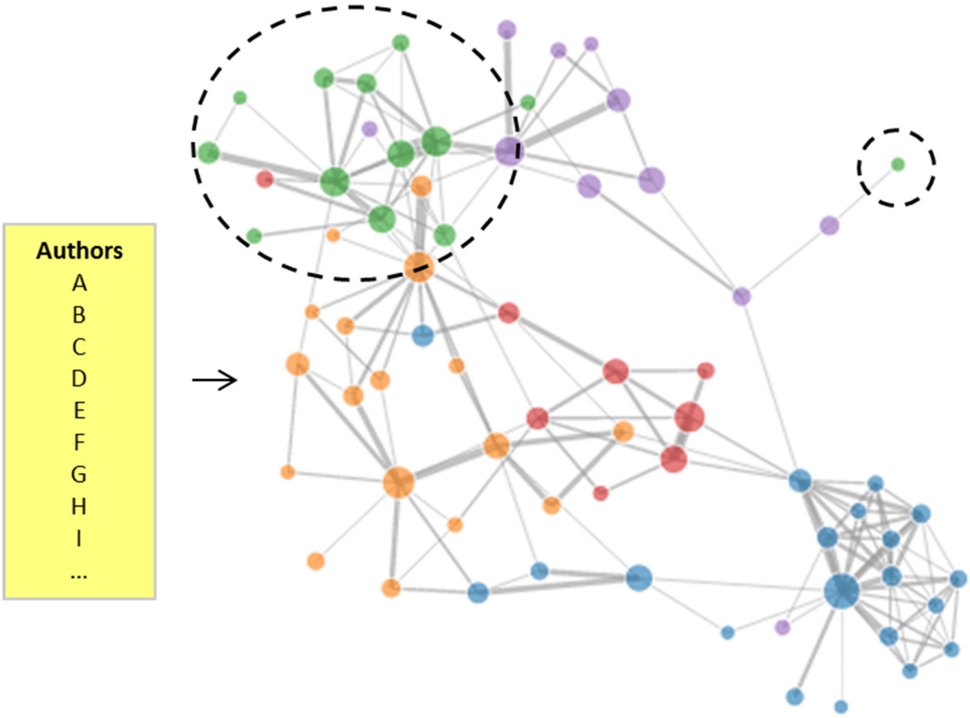


Figure 33 Structural behaviour characterisation

Structural behaviour characterisation: involves describing the configuration of connections between a set of graph elements. For example, this could be in terms of a particular local connectional pattern such as a cluster, clique, connected component, motif etc.; in general terms of the density or sparsity of connection (e.g. densely connected, tightly connected or many isolates etc.); or at a more global level view referring to the type of graph structure, such as small world, scale-free, or core/periphery network structure. For example *what is the co-authoring pattern of authors in the Computing department?* (Figure 33)

Figure 34 Structural pattern search

Structural pattern search: this is the opposite of the above task in that we seek to find the set of graph elements associated with a given pattern or configuration of connections. For example, *which authors belong to the small, densely connected cluster?* (Figure 34)

Comparison and relation seeking involving structural behaviours: Analogous to the attribute based synoptic tasks, we may also wish to compare or find relations between structural patterns, and the graph subsets associated with these patterns:

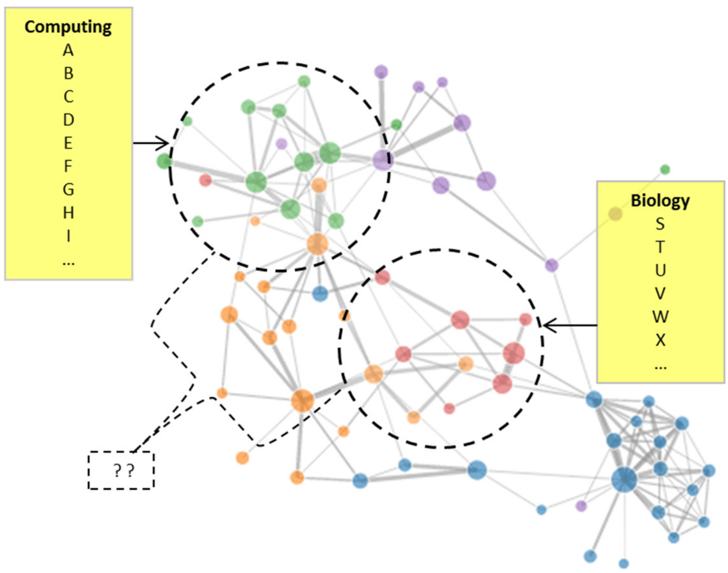


Figure 35 Direct structural comparison

Direct structural comparison: Find the relation between structural patterns (similar/different/opposite) associated with given sets of graph elements e.g. *compare the co-authoring pattern of authors in Biology with that of the Computing department.* (Figure 35)

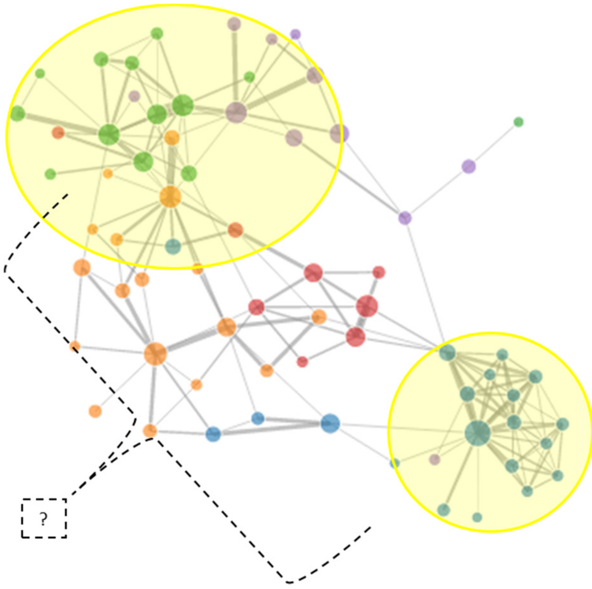


Figure 36 Inverse structural comparison

Inverse structural comparison: Find the relation between the sets of graph elements associated with given patterns (linking, distance, set relations) e.g. *how are the two largest co-authoring clusters related?* (Figure 36)

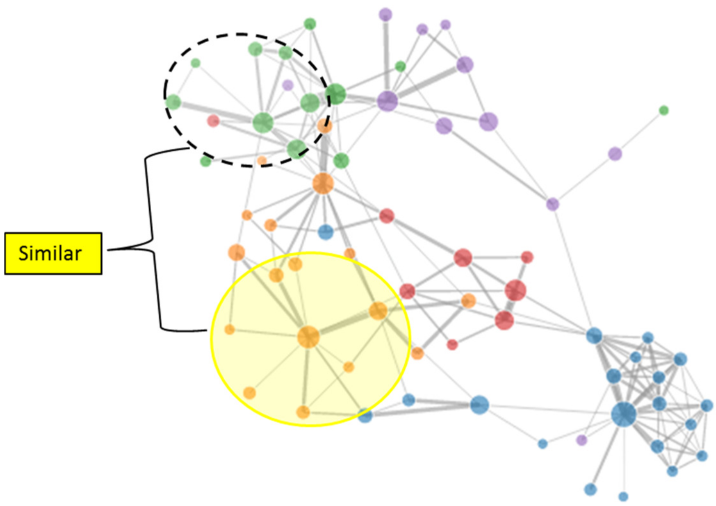


Figure 37 Structural relation seeking

Structural relation seeking: Find structural patterns related in a given way e.g. *find instances of the same network motif, or find other co-authoring clusters similar to that of co-authoring cluster A* (Figure 37). Find subsets of graph elements associated with given patterns which are related in a given way e.g. *find closely connected co-authoring clusters.*

4.2.4 Lookup, comparison, and relation seeking on edges

One final set of tasks are those involving lookup, comparison, and relation seeking on edges. For example:

- *find co-authoring relations with a weight of 4*
- *compare the co-authoring relationship between Authors A and B, with that of Authors B and C*
- *find pairs of authors with similar co-authoring relationships.*

Performing lookup, comparison, and relation seeking tasks on *relations* does not exist within the original Andrienko task framework. However, as it is preferable not to add more task categories than necessary, for these tasks, it is suggested that the edges or paths be treated as references. In so doing, the elementary attribute based tasks of the original framework can be employed. For example:

- *find co-authoring relations with a weight of 4,* becomes an inverse lookup task i.e. we want to find the reference(s) (edge(s)) associated with a given characteristic value (weight of 4).
- *compare the co-authoring relationship between Authors A and B, and that of Authors B and C,* becomes a direct comparison task i.e. we first find the co-authoring relationships (expressed in terms of e.g. existence, strength etc.), then find the relation between the them (expressed in terms of e.g. similarity/difference in existence, less than/greater than in strength etc.)
- *find pairs of authors with similar co-authoring relationships*, becomes a relation seeking task i.e. we want to find pairs of authors where the relation between them is that of *similarity* in co-authoring relationship.

4.2.5 Implications for the connection discovery tasks

So far we have considered the additional descriptive tasks required to support the extension of the data model to include non-fixed linking relations between references of the graph referrer. These relations also introduce additional possibilities for the set of connection discovery tasks (outlined in Section 3.4.3.4). For example, we may wish to investigate the effect of graph structure on attribute values, and vice versa; or the effect of patterns of connectivity in one part of the graph on the structural patterns of other parts of the graph. These are discussed further in Section 5.6.

4.2.6 Summary of extensions to the data model and task framework

To handle graph data, the data model is extended with a new referrer type – graph – whose elements are discrete, unordered, with distances. A new type of relation – linking – is also introduced, which exist between the elements of the graph referrer. As linking relations in the graph referrer are not fixed, structural behaviours, which are described by structural patterns, are introduced to capture variations in graph structure.

Under the extended data model, edges are treated as relations. This allows them to feature in the inverse lookup and comparison tasks of the original framework. The synoptic tasks of the original framework (which involve attribute based behaviours and patterns) can also be formulated for graph data. For tasks involving lookup, comparison, or relation seeking on edges, it is suggested that the edge or path be treated as a reference, and the tasks be formulated according to the original framework.

Two extensions are made to the task framework. These tasks involve only the referential components and relations between them:

- The pure relational tasks described in the Andrienko framework are instantiated to produce structural comparison (find the relations between two graph objects) and structural relation seeking tasks (find graph objects related in the given way, one of which may be specified). Variations of these tasks can be constructed according to whether nodes and/or graph objects are involved, and the relation of interest. We can think of these tasks as elementary structural tasks, as they involve relations between individual graph objects.

- A set of synoptic tasks analogous to those of the existing task framework but involving structural behaviours and patterns, are added: structural behaviour characterisation, structural pattern search, and comparison and relation seeking tasks involving structural patterns and the associated sets of nodes.

The extended framework is summarised in Figure 38.

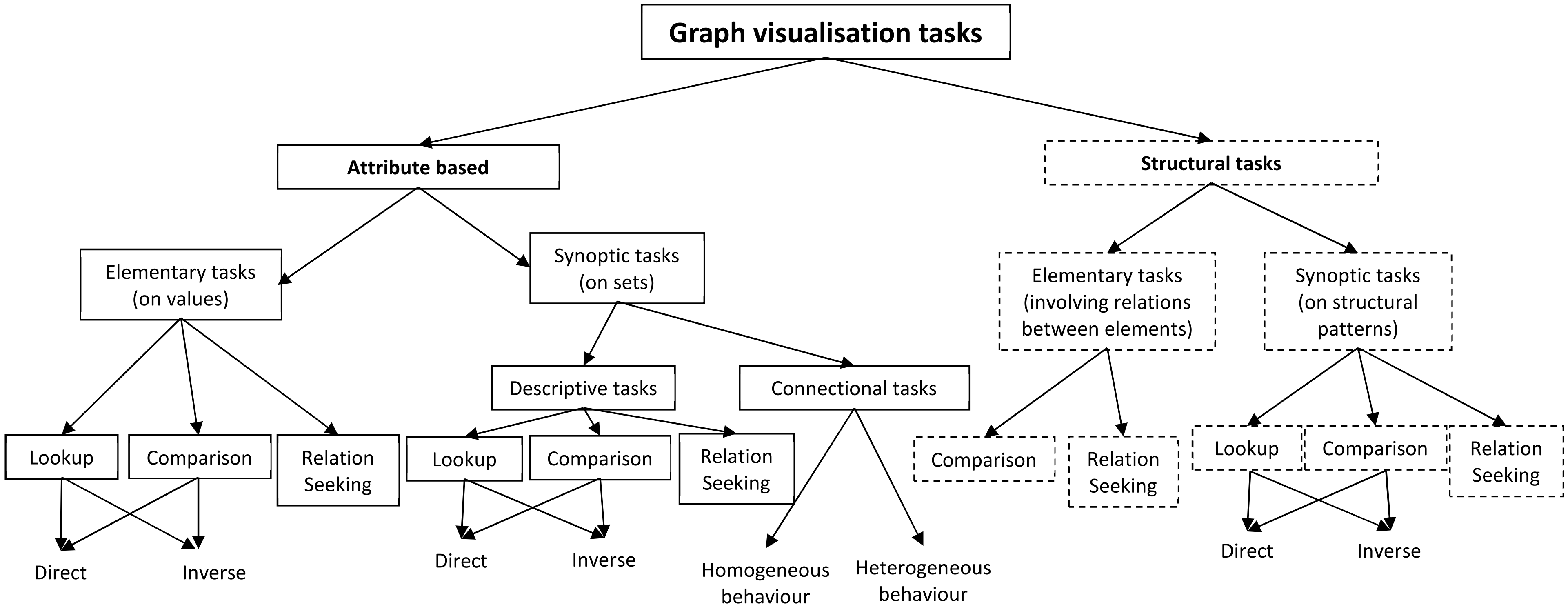


Figure 38 The extended task model. Based on Aigner et al.'s ([27] p74) drawing of the Andrienko task model organised into a taxonomy, redrawn and extended to include structural tasks for graph visualisation. (Extensions indicated with dashed lines)

4.3 The extended task framework for graph data

Table 12 and Table 13 show the tasks of the Andrienko framework extended for graph data.

Table 12 describes the tasks of the original framework applied to graphs and the additional set of synoptic tasks involving structural behaviours and patterns. The task descriptions in bold are either the Andrienko descriptions given in [5] (Table 3.5, pp.121-4), or adaptations of these. This is followed by a short explanation applying the task to graph data, and then an example task. In order to help show the differences between tasks, all of the example tasks are drawn from the same domain (an author publications data set). However, while it was possible to construct reasonable examples for each task for illustration purposes, some tasks might be more meaningful when applied to another data domain. Note that the task examples are intentionally constructed with static graphs in mind, but examples involving time are used for task variations explicitly involving multiple referrers; tasks for temporal graphs are the subject of Chapter 5.

Many of the tasks are not what we might typically think of as "graph tasks". This could perhaps be due to the focus on attributes in the original framework, and that several tasks in the resulting extended framework do not include any reference to the graph context (for example, elementary direct comparison tasks simply involve comparison of attribute values). Graph attributes also tend to be neglected more generally in the literature, with the main focus of graph visualisation papers being on how to represent graph structures. However, all of these tasks have the potential to be of interest when exploring graph data.

Another point to note is that it is clear that there are further possible variations of each task when applied to a concrete data set. For example, where there are multiple attributes, the direct comparison task involving different attributes could be formulated for every pair of comparable attributes in the data set. The synoptic tasks can be constructed to involve different types of graph objects of interest e.g. whether we are interested in clusters or paths, depending on the data set (paths might be of more interest when considering routes in transportation networks, while clusters may be of interest when studying communities in social networks). Moreover, even in abstract terms, slightly different comparison and relation seeking tasks can be formulated depending on the types of relations that exist between elements. For example, as seen in Section 4.2.2, quite different versions of the elementary structural comparison task can be constructed depending on the type of relation in which we are interested. These tasks can also be formulated with a specific type of relation in mind - *do subgraphs A and B overlap?* (set) or *are subgraphs A and B connected?* (linking) – or more generally – *in what way are subgraphs A and B related?* (i.e. set and/or linking relations could be referred to when answering this question). We will also see in Chapter 5 that when we consider multiple referrers, the number of task variations increases several fold. While it is not necessary to specify every variation of every task, it is important to bear these possibilities in mind when considering which tools are able to support which tasks, in Chapter 8.

Table 12 The elementary and synoptic descriptive tasks of the original framework are instantiated for the graph case. The final column describes additional synoptic tasks of the extended framework, which involve the structural behaviours and patterns of the extended data model. Tasks with a yellow background indicate an extension: they are either tasks of the extended task framework, or are formulated to involve the linking relations of the extended data model as a target or constraint.

	Original Framework[6]		Synoptic tasks involving structural behaviours and patterns[7]
	Elementary tasks[8]	**Synoptic tasks**	
Lookup	**Direct lookup**: find the attribute value of a given node *How many publications has Author A?*	**Behaviour characterisation (pattern definition)**: find a pattern to describe the behaviour of an attribute over the graph (or a subset of the graph) *Describe the distribution of publication counts over the co-authorship network.*	**Structural behaviour characterisation (pattern definition)**: find a pattern to describe the configuration of connections between a set of graph elements, such as a particular motif or graph structure *NB in the example tasks, author group A and author group B are used as shorthand to represent two subs sets of authors {A, B, C, D, E, F} and {G, H, I, J, K, L}, respectively.* *What is the co-authoring pattern of author group A?*
	Inverse lookup: find nodes with the given attribute value *Find authors with more than five publications.*	**Pattern search**: find the subset of nodes (graph object) corresponding to a given pattern of attribute values	**Structural pattern search:** find the set of graph elements associated with a given pattern of connections. *Which authors belong to the small densely connected cluster?*

[6] Task descriptions in bold are those given in Table 3.5 of [5], pp.121-4.
[7] Task descriptions in bold are adapted from those given for synoptic tasks in the original framework, as per Footnote 6.
[8] As noted in Section 4.2.4 these tasks can also be formulated to involve edges as references.

		Who are the authors belonging to the co-authoring cluster with very high numbers of publications?	
Direct comparison	***with specified attribute values***: find the attribute value of a given node and compare it with a given value. *Compare Author A's publication count with the average number of publications (five).*	***with a specified pattern :*** one of the patterns is specified, while the other results from a behaviour characterisation task. *Compare the pattern of publication counts over co-authoring group A, with a typical pattern (e.g. where more central authors have higher numbers of publications)*	***with a specified pattern :*** one of the patterns is specified, while the other results from a behaviour characterisation task. *Compare the co-authoring pattern of author group A with a typical co-authoring pattern.*
	between values of the same attribute(s) for different references: find and compare the attribute values of two nodes. *Compare the publication counts of Author A and Author B.* *Compare Author A's publication count in 2013 and 2014.*	***between behaviours of the same attribute(s) over different reference sets:*** find two patterns (associated with the same attribute) corresponding to two different specified graph objects, and compare them. *Compare the distribution of publication counts over co-author groups A and B.* *Compare the distribution of publication counts over co-author group A in 2012 and 2014.*	***between structural behaviours over different reference sets:*** find two patterns corresponding to two different specified graph subset, and compare them. *Compare the co-authoring pattern of author group A with that of author group B.* *Compare the co-authoring patterns of author group A in 2012 and 2014.*
	between values of different attributes for the same reference: find and compare two different attribute values of the same node.	***between behaviours of different attributes over the same reference set:*** in this case the	***between different types of structural behaviour over the same reference set:***

	Compare the number of journal articles and conference proceedings which Author A has published.	set of nodes is the same, but the behaviours of interest are those of different attributes. *Compare the distributions of journal article counts and conference proceeding counts for co-author group A.*	(only applicable where different types of relations exist in the graph, such as if we modelled a friendship network alongside the co-authorship network) *Compare the co-authoring pattern of author group A with their pattern of friendship connections.*
	between values of different attributes for (partly) different references: find and compare two different attribute values of two different nodes. *Compare the number of journal articles published by Author A with the number of conference proceedings published by Author B.*	***between behaviours of different attributes over (partly) different reference sets:*** find and compare the patterns associated with two different attributes of two different specified graph objects. *Compare the distribution of journal article counts over co-author group A with the distribution of conference proceeding counts over co-author group B.*	***between different types of structural behaviour over (partly) different reference sets:*** as above, but involving different subsets of the graph. *Compare the co-authoring pattern of author group A with the friendship pattern of author group B.* *Compare the co-authoring pattern of author group A in 2013 with the pattern of friendship connections amongst the same authors in 2012.*
Inverse comparison	***with specified reference(s)*****:** find a node with the given attribute value and compare* it with a given node. *Does Author A co-author with the author with the most publications?*	***with specified reference sets:*** compare the set of nodes resulting from a pattern search task with a specified node or set of nodes. *Does Author A belong to the co-authoring group with particularly high publications counts?*	***with specified reference sets:*** compare the set of nodes resulting from a structural pattern search task with a specified node or set of nodes.

		In what way is the co-authoring group with particularly high publications counts related to cluster A?	*In what way is Author A related to the authors belonging to the small densely connected cluster?* *In what way is author group A related to the authors of the largest co-authoring cluster?*
	between references corresponding to different values of the same attribute(s): find and compare* nodes corresponding to given attribute values; in this case the inverse lookup tasks involve the same attributes. *Do the authors with the highest and lowest publication counts co-author?*	***between the reference sets corresponding to specified behaviours of the same attribute(s)***: here the same attribute is involved in both pattern search tasks. *In what way is the area of the co-author network with particularly high publication counts related to the area with low publication counts?* *In what way is the area of the co-author network with authors predominantly belonging to the Biology department related to the area with authors mainly belonging to Computing?*	***between the reference sets corresponding to specified structural behaviours***: compare the sets of nodes resulting from two pattern search tasks. *In what way are the authors of cluster A and cluster B related?* *In what way are paths A and B related?*
	between references corresponding to specific values of different attributes: find and compare* nodes corresponding to given attribute values; in this case the inverse lookup tasks involve two different attributes.	***between the reference sets corresponding to specified behaviours of different attributes:*** here a different attribute is involved in both pattern search tasks. *In what way is the area of the co-author network with particularly high journal article*	***between the reference sets corresponding to specified structural behaviours of different types:*** (only applicable where different types of relations exist in the graph) In what way are the authors of co-authoring cluster A related to friendship cluster B?

	In what way is the author with the highest count of journal articles related to the author with the highest count of conference proceedings?	*counts related to the area with high conference proceedings counts?*	
Relation-seeking	***between values of attribute(s) and, at the same time, between references:*** both the attribute values and nodes are constrained by specified relations. *Find co-authors with similar numbers of publications.*	***between behaviours of attribute(s) and, at the same time, between reference sets:*** a relation between behaviours and a relation between graph subsets is specified; we want to find the graph subsets that are related in this way. *Find clusters of co-authors with similar distributions of publication counts, that are connected to one another.*	***between structural behaviours and, at the same time, between reference sets:*** both a relation between structural behaviours and a relation between graph subsets is specified; we want to find the graph subsets that are related in this way. *Find co-authoring clusters that are connected to one another* *Find paths that cross.*
	between characteristic(s) of a specified reference and characteristics of other references: a relation between characteristics, and one of the nodes is specified; the other node(s) must be found using a lookup task. *Find authors with more publications than author A.*	***between an attribute behaviour over a specified reference subset and attribute behaviours over other reference subsets:*** in this case a graph subset and the relation between behaviours is given; we want to find the graph subset which has a behaviour related to the behaviour of the given graph subset, in the given way. *Find clusters of co-authors with distributions of publication counts similar to that of co-author group A.*	***between a structural behaviours of a specified reference subset and structural behaviour of other reference subsets:*** in this case a graph subset and the relation between structural behaviours is specified; we want to find the graph subset which has a structural behaviour related to the behaviour of the given graph subset, in the given way. *Find sets of authors with similar patterns of connectivity to author group A.*

	***between values of the same attribute(s) for partly different references (in a dataset with multiple referrers*)**: the values of either the time or graph component are given , along with the relation between characteristics; the target is the unknown time or graph. (Note that Andrienko and Andrienko's formal description implies that the unknown reference is the same in both lookup tasks.) *Which authors had fewer publications in 2013 than in 2014?*	***between behaviours of the same attribute(s) over partly different reference sets (in a dataset with multiple referrers)***: the values of either the time or graph component are given, along with the relation between behaviours; the target is the unknown time or graph component. *Which co-authoring groups in the network had a substantial change in publication rates between 2013 and 2014?*	***between structural behaviours over partly different reference sets (in a dataset with multiple referrers)***: the values of either the time or graph component are given, along with the relation between behaviours; the target is the unknown time or graph component. *Which co-authoring groups in the network had a substantial change in their patterns of co-authorship between 2013-2014?*
	between values of different attributes for the same reference: find the node(s) with attribute values related in the given way; the attributes involved are different. *Find authors who publish more journal papers than conference proceedings.*	***between behaviours of different attributes over the same reference set***: in this case the relation is between the behaviours of two different attributes over the same graph subset *Which co-authoring group has very different distributions of journal article counts and conference proceeding counts*?	***between different types of structural behaviours over the same reference set***: (only applicable where different types of relations exist in the graph) Find a graph subset(s) which has similar patterns of co-authorship and friendship connectivity.

***** *the term* ***'****compare' here includes finding whether/in what way the two nodes are connected, in addition to the equality relation i.e. whether or not the two nodes the same, as discussed in Section 4.2.*

Table 13 'Elementary' structural tasks involving individual relations between graph objects. These tasks extend the original Andrienko framework.

	'Elementary' structural tasks[9]
Structural comparison	How are nodes **p** and **q** (or graph objects **P** and **Q**) related? *Are authors A and B co-authors?* *Does author A belong to co-authoring cluster A?* *In what way are author groups A and B connected?*
Structural relation seeking	with an additional specified element: What node (or graph object) is related to node **p** (or graph object **P**) in the way **ρ**? *With which authors does author A co-author?* *To which co-authoring cluster does author A belong?* *Which authors belong to co-authoring cluster A?* *To which co-authoring cluster(s) is co-authoring cluster A connected?*
	What nodes (or graph objects) are related in the way **ρ**? *Which authors co-author a great deal?* *Which co-authoring clusters are connected?* *Which co-authoring clusters overlap?*

[9] Adapted from the three schemes given for "pure relational tasks" in [5], pp. 62-63, discussed in Section 4.2.1.

Chapter 5 Temporal Graph Tasks

Having extended the Andrienko framework to handle graph data, this chapter uses the extended framework to elucidate the range of possible tasks involved in exploring temporal graph data.

Firstly, a recap is given of the two dimensions on which tasks in both the original and extended frameworks are categorised - level of analysis (elementary/synoptic distinction) and task type. An additional classification of the synoptic tasks is then introduced for tasks in the temporal graph case. Combining the task dimensions produces a basic task taxonomy for temporal graphs.

Next, the sub-variations in task types are considered for the temporal graph case. A systematic approach to combining these possible sub-variations in task type with the level of analysis (elementary and three variations of synoptic task), in order to produce a task design space, is discussed. This process results in a comprehensive list of the possible temporal graph tasks.

5.1 Existing task classification

Task categories in the Andrienko framework are intentionally generic in order to be utilised with any type of data. In Section 3.4.3, the two dimensions upon which tasks in the framework are classified were discussed. The approach follows Bertin [99] in classifying tasks on the basis of the structure of the data, considering the level of analysis (the elementary/synoptic task distinction), and type of data item (referential components, characteristic components, relations) participating as either task targets or constraints, which distinguishes the main task types (lookup, comparison, relation seeking). Two additions to the second dimension were made to handle graph data: (1) simpler varieties of (inverse) comparison and relation seeking tasks were introduced (structural comparison and structural relation seeking), where neither of the referential components are found via a lookup task, and (2) an additional data item – structural behaviours – which participate in tasks in a manner equivalent to that of the attribute-based behaviours of the original framework. **Figure 39** summarises the task type dimension of the extended framework, showing how tasks are distinguished according to the different types of data items which participate as targets or constraints.

Task type		Target	Constraint
Lookup	Direct	characteristic or structural behaviour	referential
	Inverse	referential	characteristic or structural behaviour
Comparison	Direct	relation	characteristic or structural behaviour*
	Inverse	relation	referential*
Structural comparison		relation	referential
Relation seeking		characteristic or structural behaviour or referential*	relation
Structural relation seeking		referential	relation

**at least one of these components is found via a lookup task*

Figure 39 Summary of the data components participating in tasks as targets or constraints in the extended framework

5.2 Additional classification for temporal graph tasks

One further data-based distinction is now made for temporal graphs. Because temporal graphs involve two referrers, time and graph, tasks can be usefully classified according to the four possible combinations of referential components which participate: time points, time intervals, graph elements, graph subsets (**Figure 40**). This classification is essentially a sub-classification of synoptic tasks; the four classes, or "quadrants", capture the Andrienko elementary/synoptic distinction, along with three variants of synoptic tasks: elementary tasks (Q1), tasks considering graph subsets (Q2), temporal subsets (Q3), and both graph and temporal subsets (Q4). A summary of this task dimension is given in Figure 41.

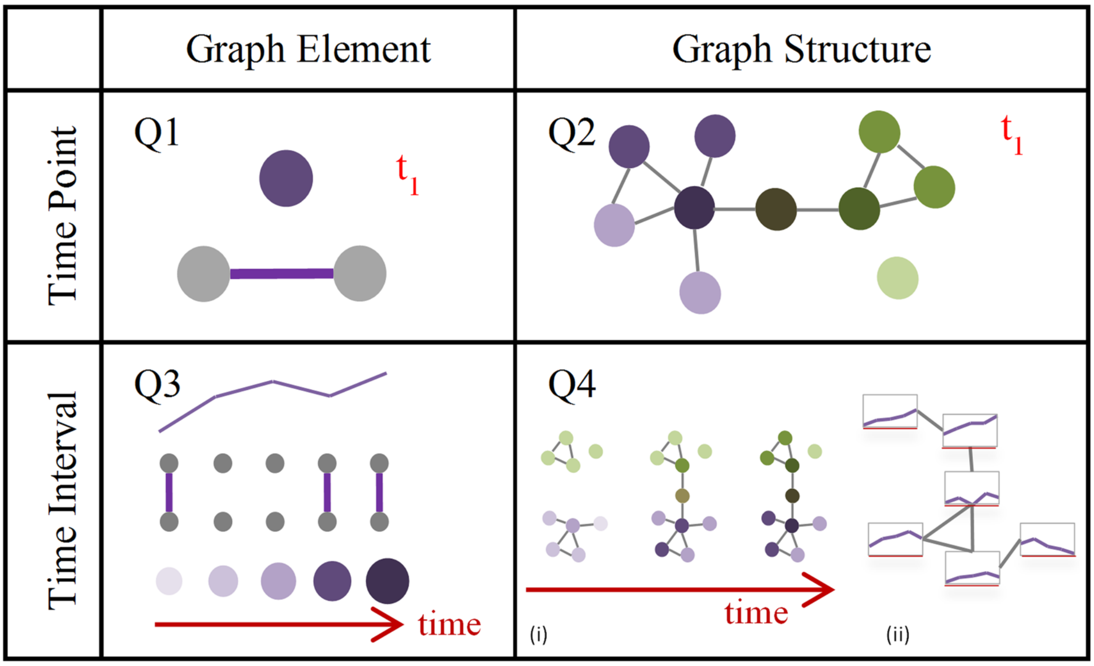


Figure 40 Four possible combinations of referential components: Q1 – individual time points and nodes or edges; Q2 – graph objects and individual time points; Q3 – individual nodes or edges and time intervals; Q4 – graph objects and time intervals.

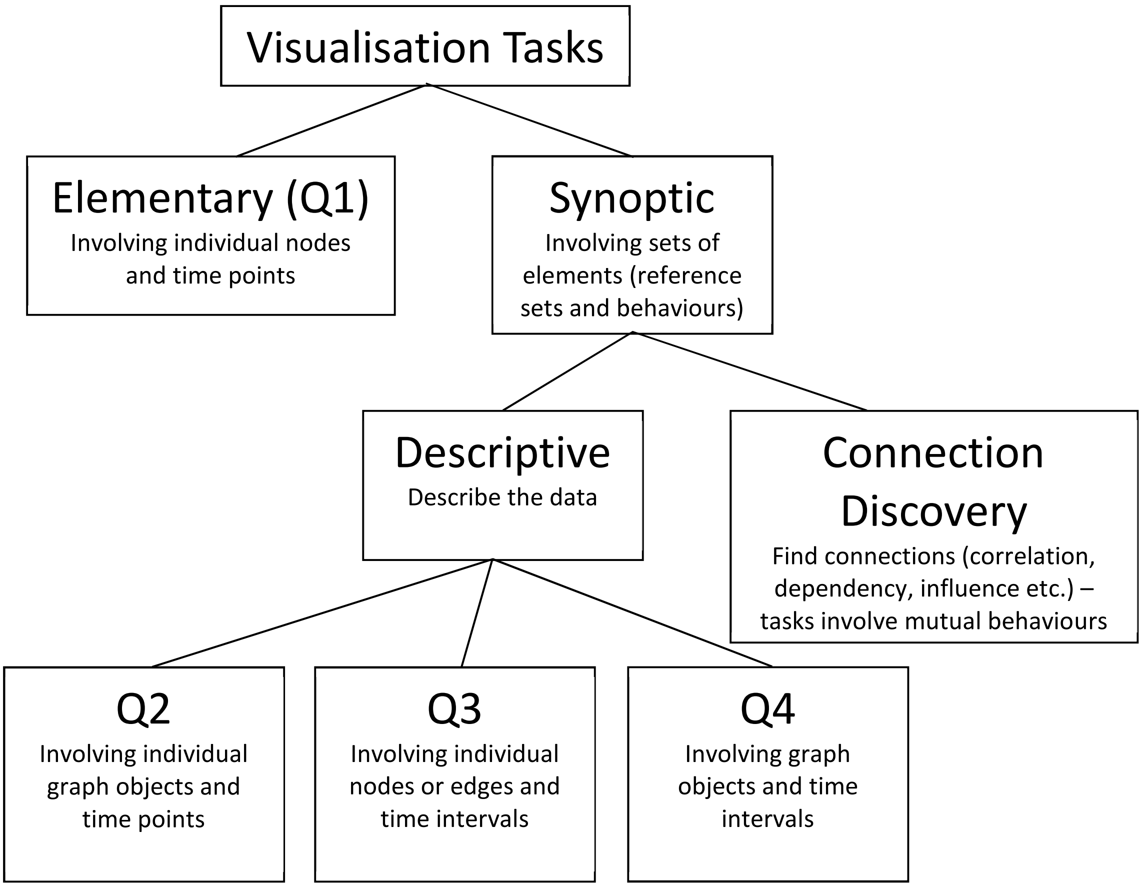


Figure 41 Summary of the Level of Analysis task dimension in the temporal graph case (note that Connection Discovery tasks are discussed separately in Section 5.6.)

Recall that one of the main purposes of the task classification in this work is to use it to consider which visual techniques are able to support which tasks. This additional classification is useful as it produces four classes of tasks which will likely require significantly different visual representations.

Each combination of referential components has different characteristic components and relations associated with it. In particular, the behaviours in Q2, Q3, and Q4, are very different. These different behaviours are now discussed.

5.2.1 Temporal graph behaviours

The synoptic tasks of the Andrienko framework play the primary role in exploratory data analysis, and behaviours are the principle notion associated with synoptic tasks ([5], p.158), which involve constructing, finding, and comparing patterns which represent these behaviours([5], p.90),. In Section 4.1.2, behaviours (and their associated patterns) for the *static* graph case were described, for example, distributions of attribute values over the graph structure. The notion of structural behaviours and patterns were also introduced, which capture the configurations of nodes depending upon the linking relations which exist between them. This section now considers the possible behaviours in the temporal graph case.

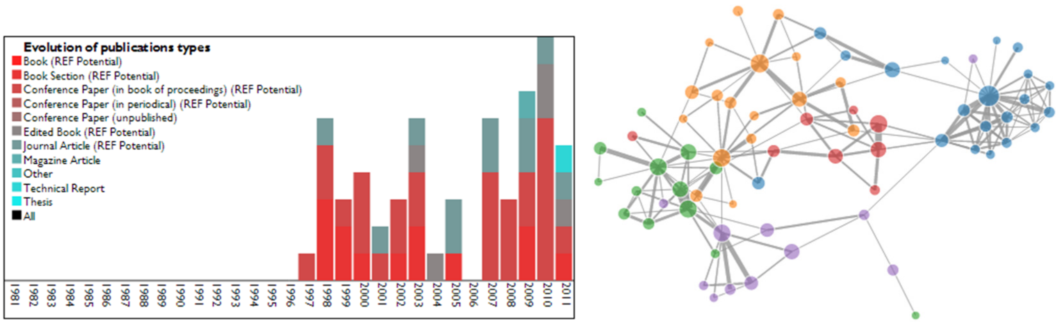


Figure 42 Partial behaviours in the temporal graph case. Left: a temporal trend in an individual author's publication count. Right: the distribution of publication counts (node size) over the co-authorship network in a specific year.

The Andrienko framework offers a detailed discussion of behaviours over multidimensional reference sets ([5], pp. 98-107). They distinguish 'partial', 'aspectual', and 'overall' behaviours. Partial behaviours are those associated with an individual reference of one of the referrers, for example, a temporal trend in an individual author's publication count, or the distribution of publication counts over the co-authorship network in a specific year (illustrated in **Figure 42**).

Aspectual behaviours consider certain aspects of the 'overall' behaviour (i.e. all behaviours over the entire data set). Where there are two referrers, there are two aspectual behaviours, each of which consider a set of partial behaviours taken together, for example, the set of temporal trends in publication counts for all authors, or the distributions of publication counts in all years. Moreover, the aspectual behaviours consider the behaviour of the partial behaviours, i.e. the distribution of the temporal trends over the network, or the temporal trends in the distributions over the network, over time (illustrated in **Figure 43**).

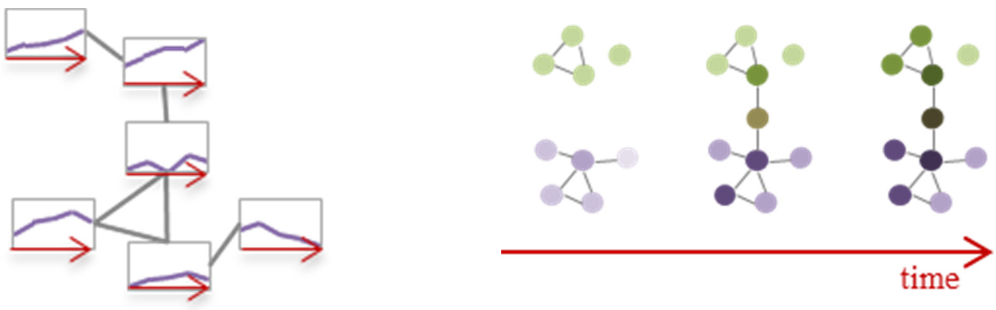


Figure 43 Aspectual behaviours. Left: distribution of the temporal trends over the network. Right: temporal trends in attribute distributions over the network, over time.

Andrienko and Andrienko demonstrates that two aspectual behaviours are not equal to one another: this can be clearly seen in the examples given above. They also stress that neither aspectual behaviour is the same as the overall behaviour, thus we obtain only partial understanding of the overall behaviour and underlying phenomena through their study, and additional effort is required to piece together these partial understandings in order to comprehend the whole.

The structural behaviours introduced in the extended framework are modelled on the original attribute-based notion of behaviours, thus we can consider analogous partial and aspectual structural behaviours. In the temporal graph case, there are therefore eight behaviours in total, two partial and two aspectual attribute based behaviours (A), and two partial and two aspectual structural behaviours (S). These can be considered according to the quadrant with which they are associated (illustrated in Figure 44):

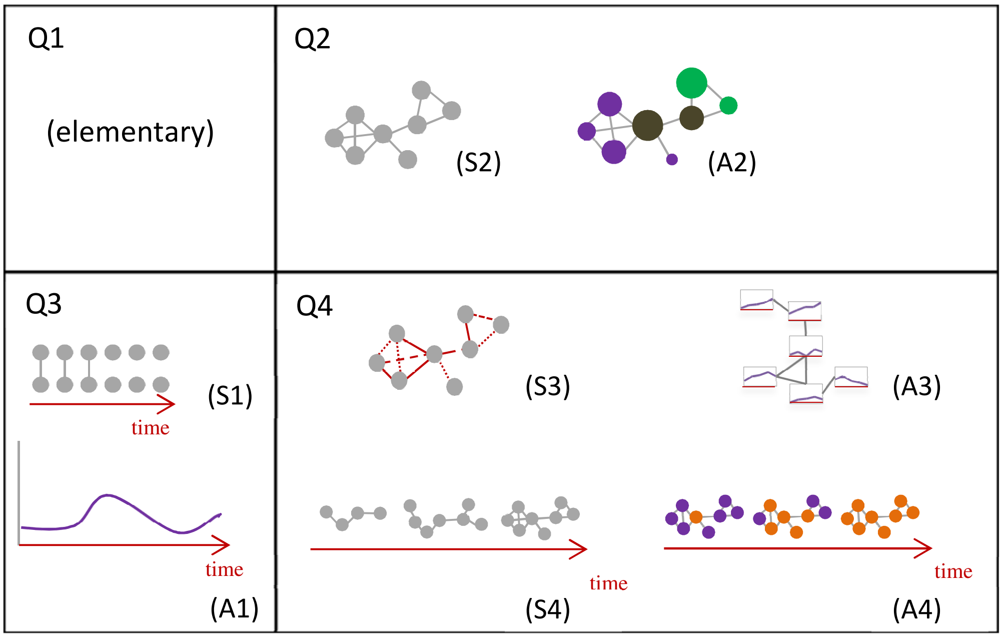


Figure 44 Illustrating the structural and attribute based behaviours in the temporal graph case, by quadrant.

Q2 involves the behaviour of an attribute over a set of nodes at a single time (A2) e.g. the distribution of an attribute value (such as publication count) over the network; and the configuration of nodes based on the linking relations between them, at a single time (S2) e.g. clusters, cliques, motifs, co-authoring groups.

Q3 involves the behaviour of an attribute of an individual graph element (a node, edge, or graph object) over time (A1) e.g. a temporal trend in the attribute of a node such as an individual author's publication count over time; and the behaviour of linking relations between two graph elements over time (S1) e.g. the pattern of change in connectivity between two nodes over time, such as the temporal pattern of co-authorship between two authors.

Q4 has four possible behaviours associated with it:

> (A3) the behaviour of the temporal trends (described by A1) distributed over the graph e.g. the distribution of individual temporal trends in author publication counts, over the graph.

(A4) the behaviour of the distribution of the attribute values over the graph (as in A2), over time e.g. the change in distribution of research group affiliation over the co-authorship network, over time.

(S3) the behaviour of the collection of behaviours in S1 i.e. the aggregate pattern of all linking relations between pairs of graph objects over time, or the distribution of individual temporal behaviours over the graph e.g. the distribution of temporal trends in co-authorship between pairs of authors, over the network.

(S4) the configurations of nodes (i.e. S2), over time e.g. the evolution of the structure of the co-authorship network over time.

5.2.1.1 Patterns in the temporal graph case

The previous section considered the possible behaviours associated with temporal graphs. This section considers in more detail the potential patterns which we may be interested in when analysing such data. As outlined in Section 3.4.2, patterns in the Andrienko framework are subjective constructs which result from an observation of a behaviour, and offer a descriptive summary of its essential features. A number of properties of patterns are outlined in the framework ([20] p90), including:

- the degree of simplification
- level of precision
- coverage of the reference set (complete or partial), and
- the presence or absence of an overlap between sub-patterns

along with four basic variants of pattern. Before considering these pattern variants and how they might apply to the temporal graph case, let us first review the discussions of behaviours and patterns of interest in the temporal graph literature.

At a very basic level, Shannon et al. [100] state that the changes that can occur in a graph include:

- nodes being added or removed
- edges being added or removed

- changes in the weight of either nodes or edges
- changes in the explicit clustering of nodes

As the attribute based behaviours described in the previous section show, change in attribute distributions over the graph should also be added to this list.

Asur et al. [101] characterise behavioural patterns of individual nodes and of communities (clusters), in temporal graphs. They consider the changes which may be undergone between two consecutive time points:

Between any two consecutive time points, clusters may…

1. Continue
2. Merge
3. Split
4. Form
5. Dissolve

Nodes may…

1. Appear
2. Disappear
3. Join a cluster
4. Leave a cluster

We could add to their list of changes in clusters that they might grow or shrink (i.e. increase/decrease in number of nodes) or become more or less connected (i.e. increase/decrease in number of edges).

Yi et al. [70] discuss change at three levels: node/dyad level, subgroup level, and global network level. In addition to the considerations so far discussed, they consider changes in a node's centrality and its positions and role in the network[10].

[10] Examples given for roles and positions include star, liaison, brokerage, gatekeeper, isolate; note that Yi et al. are specifically interested in Social Network Analysis, and these terms are most applicable in this domain. However, roles are applicable across domains, for example, Lee et al. [40] consider articulation points, which are nodes whose removal disconnects the graph.

	Structure	Attribute
Nodes	Addition Deletion Joining a cluster Leaving a cluster Change in centrality Change in positions and roles	Change in attribute value
Edge	Addition Deletion Change in role (e.g. bridge)	Change in edge weight
Cluster	Continue Merge Split Form Dissolve Grow Shrink Increase in connectivity Decrease in connectivity Change in topological structure	Change in attribute distributions and/or attribute values[11]
Path	Form Dissolve Split Merge Increase in length Decrease in length Re-ordering of nodes	Change in attribute distribution along the path and/or attribute values.
Network	Grow Shrink Increase in connectivity Decrease in connectivity Change in topological structure	Change in attribute distributions and/or attribute values

Table 14 Some possible changes in a graph according to graph object and structure or attribute change

While not considering change in a graph, Lee et al. [40] consider the role that an individual edge may play as a bridge, which is a link whose removal disconnects a graph. Yi et al. also discuss change at the global level i.e. in the overall network topology, for example, particular types of network structure may emerge, such as a core-periphery structure, multiple clusters, small world structure, or scale free

[11] attribute values may increase across the graph, but the distribution may remain the same e.g. high values in the centre of the network, lower values at the periphery

network. Similarly, we could extend this idea to clusters, which may evolve into particular topologies, such as specific network motifs.

Finally, we can take some of these ideas and apply them to paths. Paths between two nodes may form or dissolve, change in length through the addition or removal of nodes, or be re-routed (re-ordered), for example, where the nodes in the path are the same but are connected in a different order. We could also consider some of the notions associated with clusters, for example, a path may split (become disconnected) or two paths may merge.

These considerations are combined and summarised in Table 14.

So far we have largely considered the changes that may occur in a graph between two time points. Let us now consider patterns which may occur over a time interval.

Ahn et al. [41] suggest five 'shapes of change' which focus on temporal patterns of 'entity properties'[12]. Under their framework, 'entities' are nodes or dyads, subgroups of the network, or the entire network. 'Properties' are divided into structural properties (including structural metrics such as degree, centrality, modularity, transitivity etc.[13]) and domain properties (which are independent of the network structure) :

1. ***Growth or Contraction*** – *These can show whether an entity property increases or decreases over time (e.g., a community's average number of posts per member per month). It can also be aggregated from temporal features of multiple individual events. For example, the network growth might be defined as the number of node/link additions per month. They typically involve counts and statistics.*
2. ***Convergence or Divergence*** – *A property can grow or contract during its initial stage but gradually becomes stable. Conversely, a stable property can become unstable.*

[12] Note that they capture node and edge addition/deletion and the formation and dissolution of clusters, paths etc. separately, as "temporal features of individual events".

[13] It is not clear whether they intend to also include the actual topology of an "entity", or just a metric which can describe it – the shapes of change would suggest the latter, as they are most applicable to individual values over time.

3. ***Stability*** *– There is no or little change over time.*
4. ***Repetition*** *– The repetition of specific patterns over time. It can fluctuate or show ritual behaviours.*
5. ***Peak or Valley*** *– Whether an entity property increases or decreases abruptly and then returns to its earlier value.*

(Ahn et al. [41] Section 4.5.2)

The main difficulty for Ahn et al.'s list of patterns is that they are largely applicable to patterns of numeric values over time. For example, we could describe the change in a centrality metric associated with a node over time as growing, repeating, or peaking, but we would have difficulty capturing the changes in topological structures (such as a merging of two clusters or emergence of a particular network structure), or change in attribute distributions over the graph over time (such as the spread of an attribute value from the central nodes of the network outwards), using these patterns. In order to investigate the potential behaviours and patterns of interest in the temporal graph case, the four variants of pattern discussed in the Andrienko framework are considered for the temporal graph case.

5.2.1.2 Patterns in the Andrienko framework

Andrienko and Andrienko identify four basic variants of pattern (note that these variants are not intended to be an exhaustive list):

1. ***Association****: Perception or description of a (sub)set of references as a unified whole on the basis of similarity of their characteristics, i.e. close values of one or more attributes corresponding to these references.*

2. ***Differentiation****: Perception or description of some references or subsets of references as differing from others by to their characteristics.*

3. ***Arrangement****: An idea or description of how characteristics are arranged, with respect to an ordering of references, for example a trend in characteristic that changes over time.*

4. ***Distribution summary**: A general idea or description of how characteristics are distributed over a reference set: how varied they are, what values occur most frequently, whether there are outliers (a few values greatly differing from the rest), etc.*

(Andrienko and Andrienko, [5], p91)

To illustrate, an example of an association pattern is an area with similar attribute values in space, or periods of similar attribute values in time, for example a time period during which Author A had consistently high levels of publications. Differentiation describes outliers, for example, a particular time point with a very high or low attribute value (‘*Author A had a highly productive year in 2008’*), or a location with a very different value from those by which it is surrounded. Arrangement patterns involve order – either a natural ordering, such as in time, which results in temporal trends in attribute values - or introduced, for example, if we organised the set of authors in our publication data by number of journal articles published, we might see a corresponding trend in conference papers published. Examples of distribution summaries include averages (‘*the average number of publications in 2010 was 4*’; ‘*in general the authors’ publication counts are increasing*’), distributions over space (‘*high values in the north and low values in the south*’) or frequency distributions (the count of authors belonging to different subject areas).

Table 15 and Table 16 show how these patterns could be applied to the graph case, in each quadrant, both in terms of patterns of attributes and structural patterns. Like Andrienko and Andrienko, an exhaustive list of patterns is not sought here; the purpose of this section is to give an idea of what might be of interest in each quadrant, by pattern type. Note that many of the examples below are based on those given in Andrienko and Andrienko [5], Table 3.3., p98.

Structural	Q2 (graph)	Q3 (time)	Q4 (graph over time)	Q4 (time over graph)
Association	clusters of tightly connected components, motifs	a period of connectivity or disconnection between the nodes	a period in which the graph structure/a particular structural pattern is stable	a cluster (or group) of dyads with very similar patterns of connectivity over time.
Differentiation	isolates, disconnected components	a brief period of connectivity within a longer period of disconnection; a period of extreme variation in connectivity	a time point at which a particular structural pattern occurs which is very different to those of the time period within which it lies; a period of highly changeable structural patterns.	a dyad with high connectivity over time in an area of the graph with low connectivity over time
Arrangement	with respect to the ordering of nodes by one structural metric e.g. degree, the pattern of another structural metric e.g. centrality ('centrality increases with degree')	a period of alternating connectivity and disconnection	a period in which structural patterns form or dissolve; a period of alternating structural patterns.	'dyad connectivity over time becomes less stable as we move toward the periphery of the graph'
Distribution summary	graph level statistical metrics (size, density, number of connected components etc.); frequency distribution of node/edge based statistics; position of clusters/motifs within the graph ('7-node cliques are found toward the centre of the graph')	'nodes were connected at the beginning of the time period, intermittently connected in the middle, and disconnected toward the end of the time period'	'at the beginning of the time period the graph is loosely connected; a number of clusters begin to form, which, by the end of the time period, have joined together to create a connected graph structure'	'dyads in the centre of the graph tend to be connected over the entire period, while those at the edges are more intermittently connected, with few connected towards the end of the time period'

Table 15 Examples of Andrienko's four variants of pattern applied in the graph structural case

Attribute-based	Q2 (graph)	Q3 (time)	Q4 (graph over time)	Q4 (time over graph)
Association	a cluster of nodes with high values of one attribute and low values of another	a period of high values in one attribute and low values in another; a period of relative stability.	a period of stability in a particular attribute distribution(s).	a cluster of similar temporal trends in attribute values
Differentiation	a node, or cluster of nodes, with low attribute values, within a subgraph of mostly high attribute values; a subgraph with high variability in attribute values.	a time point with a very high attribute value; a period of highly changeable attribute values.	a time point with a very different attribute distribution to those of the time period within which it lies (e.g. when the map of the market suddenly goes red); a period of highly changeable attribute distributions.	a node whose temporal trend in attribute values is opposite to/much higher/lower than those of surrounding nodes
Arrangement	'Attribute values increase toward the centre of the graph'	a period of increasing or decreasing attribute values; a period of alternating attribute values.	a temporal pattern (trend) in attribute distributions over the graph (e.g. a particular attribute value is spreading over the graph, over time)	temporal trends in attribute values increase more rapidly towards the centre of the graph
Distribution summary	'attribute values are high in the centre of the graph, and the periphery, and low to average in between'	'attribute values are low at the beginning and end of the time period, and high in the middle'	'at the beginning of the time period, attribute values are high in the centre of the graph and at the periphery, and low to average in between, but by the end of the time period, there is a move toward higher values throughout the graph'	'strongly increasing temporal trends can be seen in the centre of the graph, with nodes on the periphery having static or decreasing trends in attribute values over time'; 'temporal trends in attribute values are strongly increasing at the centre of the graph, and at the periphery, and static or decreasing in between'.

Table 16 Examples of Andrienko's four variants of pattern applied in the case of graph attributes.

Note that in the graph case (Q2), we can also consider patterns which do not involve the graph structure (i.e. if we were to treat the nodes as a population type referrer and consider them as a set of references at a single time point (in this quadrant)) e.g.

- A set of nodes with particularly high values of one attribute and low values of another, regardless of connectivity (association)
- A node with a particularly high or low attribute value (differentiation)
- With respect to the ordering of one attribute value, the pattern of another attribute value; or some combination of attribute values and structural metrics e.g. with respect to the ordering of nodes by one structural metric e.g. degree, the pattern of an attribute's values (arrangement)
- Frequency distribution of particular attribute values or structural metrics, or some aggregated metric describing all of the values in the graph e.g. total, mean/median etc. plus some measure of variance. (*distribution summary*)

In the case of graph over time (Q4), we might also be interested in these patterns over time e.g. a temporal trend in the frequency distribution of attribute values, such as an increase in the number of nodes with a particular category of value; a general shift toward lower attribute values over time, etc.

Finally, Andrienko also consider average or mean values to be patterns which are simply specified at a low level of granularity. This means, for example, that in Quadrant 4, we might be interested in the average trend in attribute values over time of all nodes, or a cumulative graph structure which shows all nodes and all edges that appear in the graph at any time, and perhaps the distribution of average node values or average edge weights over such a graph.

5.2.2 Combining the task dimensions

Applying the task types to each quadrant produces the main categories of (descriptive) tasks in the task taxonomy for temporal graphs; these are summarised in **Table 18**. It is clear that very different visual approaches will be required to carry out the same task type in each quadrant. For example, direct comparison in Q2 involves comparing attribute distributions of graphs, in Q3, comparing temporal trends in individual attribute values, and in Q4 comparing evolution of attribute distributions over the graph over time or distributions of temporal trends in attribute values over graphs (illustrated in Figure 45). This is discussed further in Chapter 8.

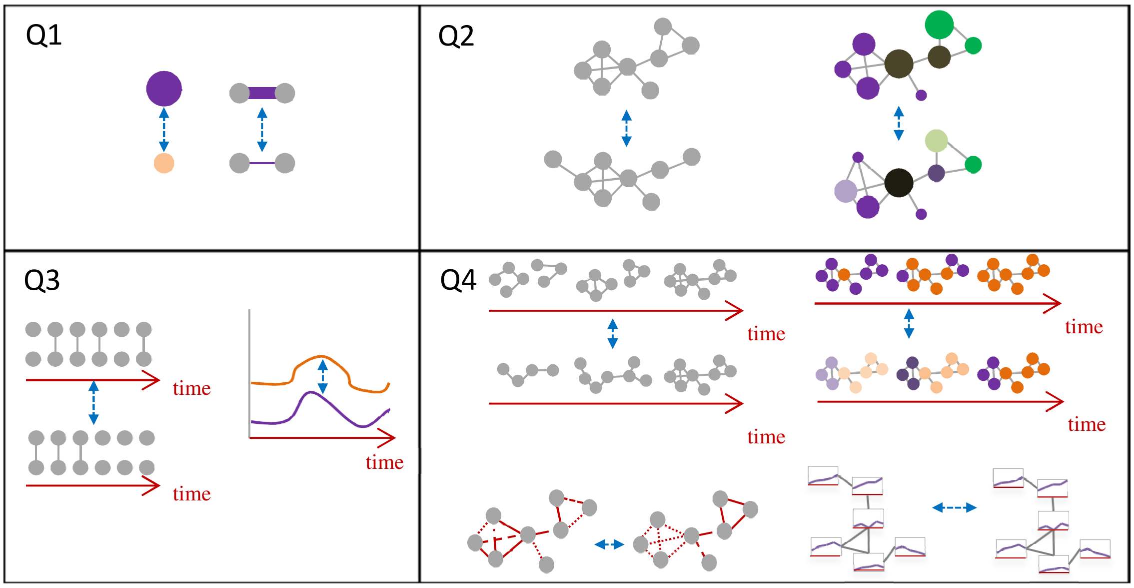


Figure 45 Differences in the direct comparison task when applied in each of the four quadrants (comparison involving structural patterns indicated to left of each quadrant in grey).

Task type		Elementary	Synoptic
Lookup	Direct	Given a time point and node[14], find its attribute value	Describe the attribute or structural pattern associated with a given time and graph component
	Inverse	Find the node(s)/ time point(s) associated with an attribute value	Find the time and graph component with a particular attribute or structural pattern
Comparison	Direct	Compare node attribute values[15]	Compare patterns of attribute values[15] Compare structural patterns[15]
	Inverse	Compare nodes/ time points[16] Find the relation[17] between nodes[18]	Compare time components[16] Compare graph components[16]
Structural comparison		Find the relation[17] between nodes (or sets of nodes)[19]	-
Relation seeking		Find attribute values (and possibly the corresponding nodes/time points) related in the given way[20] Find nodes related in a given way[20]	Find attribute or structural patterns (and possibly the corresponding graph/time components) related in a given way
Structural relation seeking		Find nodes (or sets of nodes) related in a given way	-

Table 17 Overview of elementary and synoptic tasks in the temporal graph case. The patterns participating in the synoptic tasks may describe any of the behaviours outlined in Section 5.2.1. A more detailed breakdown of tasks by quadrant is given in Table 18.

[14] Or edge, in the case where edges are treated as references (applies to all the elementary tasks)
[15] At least one of the attribute components is found via a direct lookup task
[16] At least one of which is found via an inverse lookup task
[17] Linking, distance, order, set e.g. find whether the nodes are connected
[18] At least one of the nodes is found via an inverse lookup task
[19] No inverse lookup task is involved.
[20] Involves at least one lookup task

	Q1	Q2	Q3	Q4
Lookup – direct/ behaviour characterisation	Find a node's[21] attribute value at a single time point.	Describe the pattern of attribute values associated with a set of nodes, at a single time point. Describe the structural pattern of a given set of nodes, at a single time point.	Describe the temporal trend of a node's21 attribute value. Describe the pattern of connectivity between a pair of nodes, over time.	Describe the changes in the attribute distribution over the graph, over time. Describe the distribution of temporal trends in node attributes, over the graph. Describe the changes in the structural pattern of a set of nodes, over time. Describe the distribution of temporal trends in connectivity between pairs of nodes, over the graph.
Lookup – inverse/ pattern search	Find the node(s)21/time point(s) associated with an attribute value.	Find the set(s) of nodes associated with a given attribute pattern and/or the time point(s) at which the pattern occurs. Find the set(s) of nodes associated with a given structural pattern and/or the time point(s) at which the pattern occurs.	Find the node(s) having a particular temporal trend in attribute value and/or the time period(s) over which the pattern occurs Find the node(s) having a particular pattern of connectivity and/or the time period(s) over which the pattern occurs	Find the graph (subset(s)) and/or time interval(s) over which a pattern (either attribute based or structural) occurs.

[21] Or edge, in the case where edges are treated as references

Comparison - direct	Compare node[21] attribute values [22].	Compare patterns of attribute values[22]. Compare structural patterns.	Compare temporal trends[22] in attribute values. Compare patterns of connectivity over time.	Compare patterns of attribute values[22]. Compare structural patterns.
Comparison - inverse	Compare nodes[21]/time points[23]. Find the relation[24] between nodes[23].	Compare the time points at which patterns of attribute values occur. Compare the time points at which structural patterns occur. Compare[25] the sets of nodes associated with particular attribute or structural patterns.	Compare the time periods over which patterns occur. Compare nodes having particular trends in attribute values. Compare pairs of nodes having particular patterns of connectivity over time, or the time periods over which particular patterns of connectivity occur.	Compare time intervals/graph subsets over which a particular pattern (either attribute based or structural) occurs
Structural comparison	Find the relation[24] between nodes[26].			
Relation seeking	Find attribute values (and possibly the corresponding node(s)[21]/time point(s)) related in the given way.	Find attribute patterns (and possibly the corresponding sets of nodes/time point(s)) related in a given way.	Find temporal trends in attribute values (and possibly the corresponding node(s)/time periods) which are related in a given way.	Find patterns (either attribute based or structural) related in a given way, and possibly the corresponding graph subsets/time periods.

[22] At least one of which results from a direct lookup task

[23] At least one of which results from an inverse lookup task

[24] Linking, distance, order, set e.g. find whether the nodes are connected.

[25] This includes finding the linking relations between the sets of nodes i.e. whether the sets of nodes are connected.

[26] Or sets of nodes.

	Find nodes23 related in the given way.	Find structural patterns related in a given way.	Find temporal trends in connectivity (and possibly the corresponding node(s)/time periods) which are related in a given way.	
Structural relation seeking	Find nodes26 related in the given way.			

Table 18 Combining task type and "level of analysis" (quadrants) to produce an overview of categories in the temporal graph task taxonomy.

5.3 **Sub-variations within task type**

The previous section gave a high level overview of task categories for temporal graphs, by combining the task types with the four different categories of data item which may participate in them. This section now considers the detailed variations of the main task types discussed in the Andrienko framework, and how they can be applied to tasks involving temporal graph data.

As described in Chapter 4, the Andrienko framework discusses in-depth a number of variants within the task types arising from specifying additional data items as constraints, or from particular properties of the data items participating in the task. Multiple referrers (as is the case in temporal graph data) compound the possible task variations. These variations are now discussed in the context of temporal graph data. Note that this discussion primarily relates to the attribute-based tasks of the original framework; variations in the structural tasks of the extended framework are discussed in Section 5.4.1.4.

5.3.1 Additional constraints in lookup tasks

Firstly, in lookup tasks, as we have two referrers, it is possible to construct inverse lookup tasks where one or other of the referrers is specified in addition to the characteristic. This gives three variations of inverse lookup (illustrated in **Table 19**).

Graph	Time	Characteristic	Example
✓	?	✓	*find the times at which Author A had a publication count greater than 6*
?	✓	✓	*find the authors who had a publication count greater than 6 in 2012*
?	?	✓	*find any author at any time which had a value greater than 6*

Table 19 Three variations of elementary inverse lookup in the case of temporal graphs (due to multiple referrers). Note that similar variations for the synoptic tasks can also be constructed e.g. *find the author(s) who had an increasing trend in publication count between 2010 and 2014*).

As all comparison and relation seeking tasks (at least in the original tasks of the Andrienko framework) involve at least one lookup task, it is possible to construct a wide variety of tasks simply based on the combinations of differently specified lookup tasks involved e.g. we might wish to *compare the times at which Author A had publication counts greater than 6* or *compare the authors who had publication counts greater than 6 in 2012* etc.

5.3.2 Same or different referential components

An additional variation considered in the framework in the comparison and relation seeking tasks is whether the same or different referential components are involved in the task. Again, due to the two referential components, three variations of what is meant by 'different' references are possible in the temporal graph case. These are illustrated in **Table 20**.

Graph	Time	Example
Same	Different	*compare Author A's publication count in 2012 with his publication count in 2013*
Different	Same	*compare Author A and Author B's publication counts in 2012*
Different	Different	*compare Author A's publication count in 2012 with Author B's publication count in 2013.*

Table 20 Variations in tasks based on whether the same or different referential components are involved

5.3.3 Same or different attributes

There is also the possibility of tasks involving comparison between different attributes (assuming the value domains are comparable) e.g. *compare Author A's journal publication count with his conference paper count in 2012*. In addition to this example (where time and graph components are the same in each lookup task), each of the tasks in **Table 20** can be formulated to involve either the same or different attributes in the lookup tasks, resulting in seven variations (**Table 21**).

Graph	Time	Attribute
Same	Same	Different
Same	Different	Same
Same	Different	Different
Different	Same	Same
Different	Same	Different
Different	Different	Same
Different	Different	Different

Table 21 Variations in tasks depending on the combinations of same or different referential and attribute components involved

5.3.4 Specified components

The framework also considers the possibility of comparison with some specified attribute value or referential component i.e. where only one lookup task is involved (*compare Author A's publication count in 2012 with the average number of publications (5)*).

5.3.5 Combinations in inverse comparison and relation seeking

In the case of inverse comparison, the combinations are multiplied by the three levels of specification outlined above, which may appear in a variety of combinations e.g. *compare the years in which Author A belonged to the Computing Department, and the years in which he belonged to the Biology Department* or *compare the years in which any author had a journal publication count greater than 4, and the years in which any author had a conference paper on count greater than 7*.

Similar variations can be applied to relation seeking tasks. An additional relation on the referential component (e.g. that nodes are connected, that the time periods are adjacent) can also be specified.

5.3.6 Summary

In summary, we can consider the following variations in the task types when formulating tasks involving temporal graph data:

In inverse comparison:

- Additional constraints (specified elements) in the referential component.

In comparison and relation seeking:

- Same or different graph component
- Same or different time component
- Same or different attribute
- Additional constraints (specified elements) in the inverse lookup sub-tasks (in inverse comparison).
- Additional specified relations.

5.4 **Combining the task dimensions to produce a task design space**

To recap: Section 5.2 combined two dimensions, task type and four categories of data item (quadrants) to produce a high level overview of task categories for temporal graphs. Section 5.3 considered the many possible sub-variants within the task types, which can also be combined with the four quadrants. An important point to note (which is discussed further in Chapter 8) is that the task variants within the same task type may potentially require support from quite different visual tools (for example, comparing the same attribute or two different attributes). A systematic way to investigate the possible variants of task, and a logical way to group together similar tasks (i.e. those requiring similar visual techniques for their support) was therefore sought. By systematically combining sub variations in task type with the four quadrants, a vast set of tasks for temporal graph data can be generated. In this section, the approach taken to organising the task dimensions is discussed; the full task listing can be found in Appendix A.

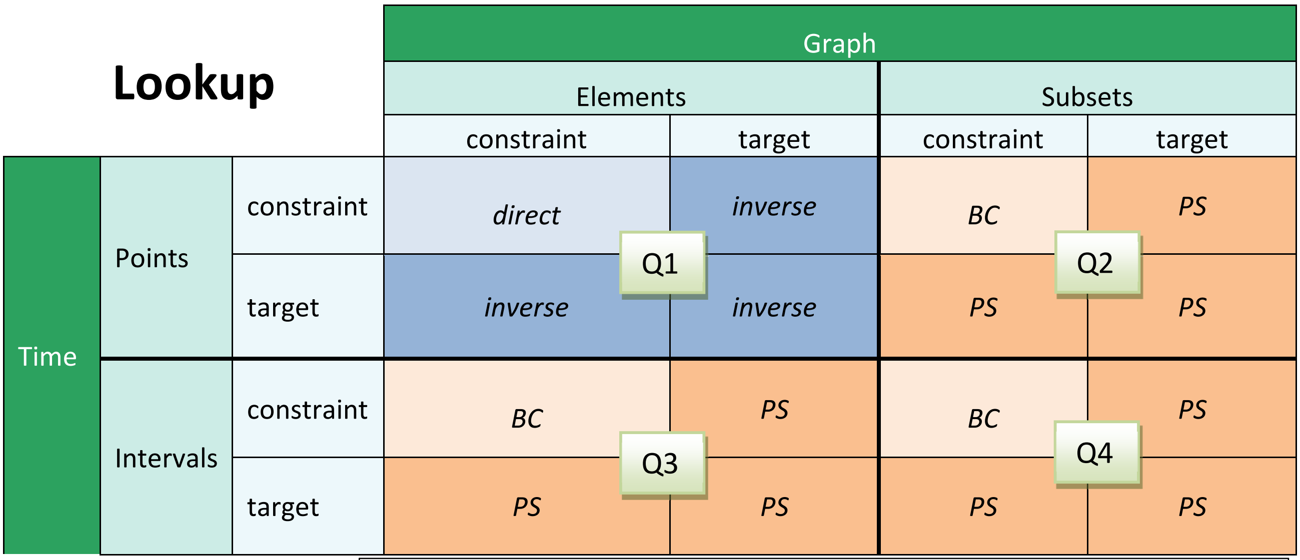
5.4.1 Task matrices

The tasks are first divided into three matrices based on the main task types (lookup, comparison, relation seeking). Each matrix is then divided into quadrants based on the referential components involved in the task: time points, time intervals, graph elements, graph subsets. This distinguishes the elementary and synoptic tasks, with elementary tasks appearing in Q1, and the three variations of synoptic tasks in Q2-4 (as illustrated in **Figure 40**). Each quadrant is then subdivided according to whether the time and graph components are specified (constraints) or unspecified (targets). This captures the inverse/direct task distinction in lookup and comparison tasks, with direct tasks appearing in the top left of each quadrant. Comparison and relation seeking with a specified component also naturally emerges where all elements of one of the lookup tasks are specified. Tasks within each quadrant move from being highly specified (top left) to most loosely specified i.e. with fewest constraints (bottom right).

In the comparison and relation seeking matrices, an additional subdivision is made relating to whether the same, or two different, temporal and/or graph components participate. The majority of tasks in these matrices can be formulated to involve the same or two different attributes (only those involving the same time and graph components cannot involve the same attribute, and these cases are noted in the task matrices).

5.4.1.1 Lookup

Figure 46 gives an overview of the lookup task matrix, which shows the variation in lookup task according to which items are specified: in Q1 there is one direct lookup task and three variations of inverse lookup, based on the combination of specified time and graph components. Similarly each of Q2-4 contains one behaviour characterisation task and three variations of pattern search.



Key to task matrix shading: Elementary tasks (blue); Synoptic tasks (orange); Direct tasks (light blue/orange); Inverse tasks (dark blue/orange); No task (grey)

Figure 46 summary of the lookup task matrix: elementary tasks appear in quadrant 1; the three variations of synoptic tasks in quadrants 2-4. In Q1 there is one direct lookup task and three variations of inverse lookup, based on the combination of specified time and graph components.

Similarly each of Q2-4 contains one behaviour characterisation task (BC) and three variations of pattern search (PS).

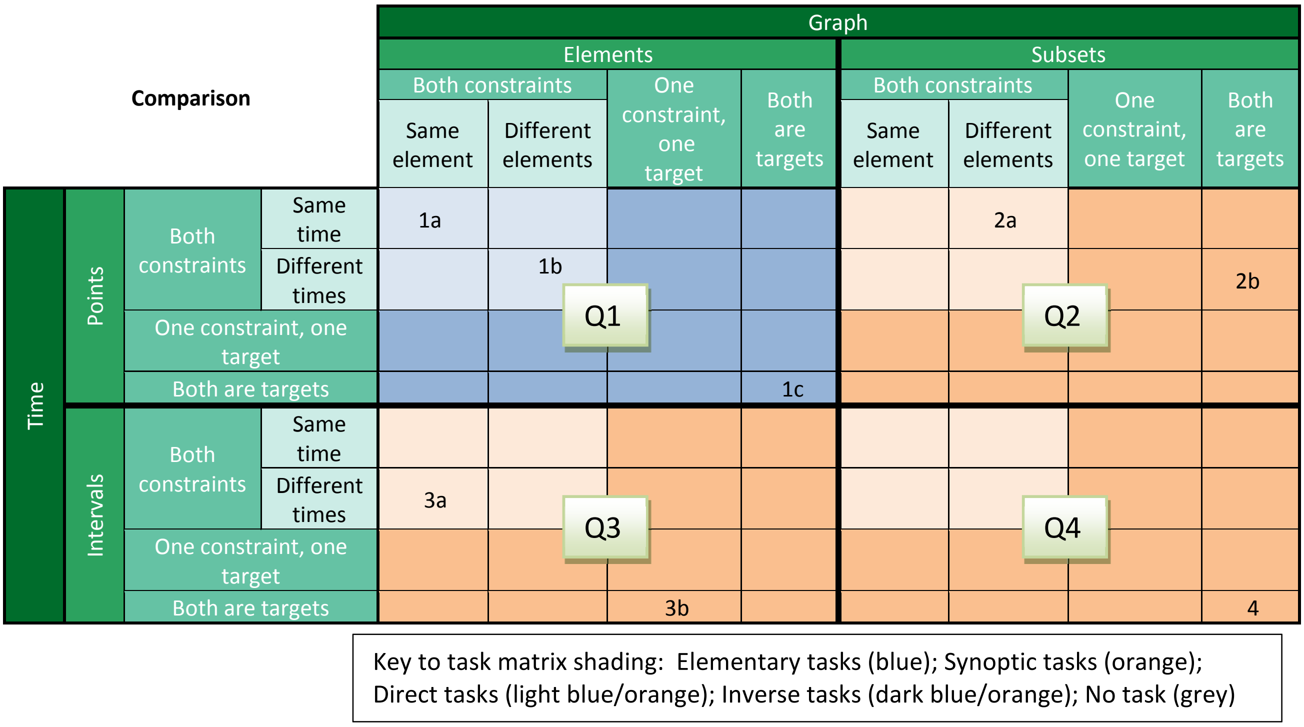
5.4.1.2 Comparison

Figure 47 gives an overview of the comparison task matrix, which shows the variations in tasks depending upon the referential components involved and whether they participate as targets or constraints. Based on this, each quadrant has 16 possible variations (4 direct and 12 inverse tasks) depending upon which, and how many, data items are specified. The task matrices do not show the variations of tasks involving the same/different attributes, but all tasks (with the exception of direct comparisons involving the same time point/interval and graph element/subset) could potentially be formulated to consider comparison involving the same attributes or two different attributes in the lookup subtasks. Additionally, where both graph and/or both time components are unspecified, tasks can be formulated with a relation as an additional constraint e.g. when comparing graph objects, the additional specified relation might be that they are the same, connected, a certain distance from one another, etc.

Figure 47 Overview of the comparison task matrix: light coloured cells in the top left of each quadrant indicate direct comparison tasks; all other cells contain inverse comparison tasks; blue and orange colours indicate elementary and synoptic tasks respectively. All tasks[27] can be formulated to involve either the same, or two different, attributes. Where both graph and/or both time components are unspecified, tasks can be formulated with a relation as an additional constraint e.g.

[27] With the exception of tasks involving the same graph and time components (top left in each quadrant), which only make sense when formulated to involve two different attributes.

when comparing graph objects, the additional specified relation might be that they are the same, connected, a certain distance from one another etc.



Example attribute-based comparison tasks:
1a. Direct comparison Compare the values of different attributes for a given node at a given time point.
1b. Direct comparison Compare the attribute values associated with two different nodes at two different times.
1c. Inverse comparison Find the time points and nodes associated with two given attribute values and compare them.
2a. Direct comparison of the attribute patterns over two different subsets of the graph at the same time point.
2b. Inverse comparison of two graph subsets associated with two given patterns at two different, specified time points.
3a. Direct comparison of the patterns of the same graph element over two different time intervals.
3b. Inverse comparison of a specified graph element and a graph element associated with a given pattern (over an unspecified time interval) and comparison of the time intervals over which the patterns occur.
4. Inverse comparison of graph subsets and time intervals associated with given patterns.

5.4.1.3 Relation Seeking

The task matrix for relation seeking (illustrated in Figure 48) is structured in an identical way to that of the comparison matrix, however the case where both graph and temporal components are specified is not applicable: the relation seeking task is to find components related in a given way, and therefore, at least one of these data items must participate as a target (i.e. an unknown item) in the task.

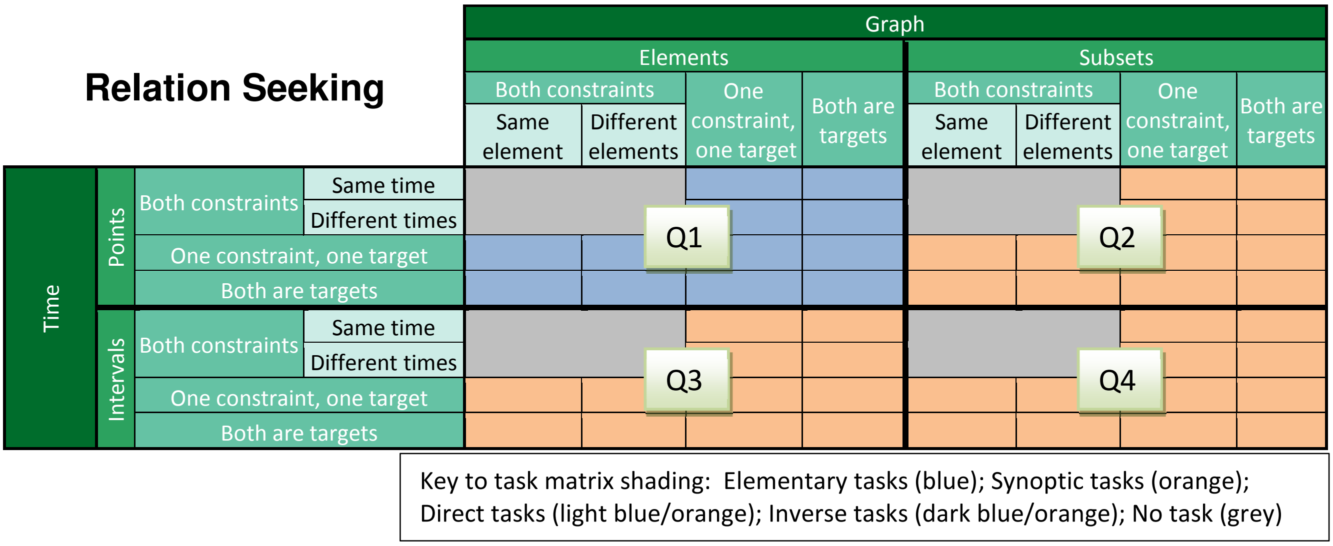


Figure 48 Relation Seeking task matrix. As for comparison tasks, with the exception that the case where both graph and temporal components are specified is not applicable (grey cells): the relation seeking task is to find components related in a given way, and therefore, at least one of these data items must participate as a target (i.e. an unknown item) in the task.

5.4.1.4 Structural Tasks

As for the attribute based tasks, the structural task space is divided based on the referential components involved. Q1 contains the elementary tasks, while the other three quadrants contain the synoptic tasks involving the partial and aspectual structural patterns. The structural elementary tasks are more limited than their attribute based counterparts: the variations of these tasks are shown in **Figure 49**. The variations of the synoptic tasks (pattern characterisation and search, comparison, and relation seeking) are directly reflective of their attribute based counterparts. However, rather than involving patterns associated with the behaviour of attribute values over reference subsets, they involve structural patterns associated with the reference subsets. The task matrices for these tasks are therefore not repeated.

		Graph elements (nodes, graph objects)		
		Both elements specified	**One element specified**	**Neither element specified**
Time points	**Both time points specified**	**Find connections between elements (comparison)** (How) is graph element $\mathbf{g_1}$ connected to graph element $\mathbf{g_2}$ at the given time, $\mathbf{t}$? $?\lambda: (\mathbf{g_1}, \mathbf{t})\ \lambda\ (\mathbf{g_2}, \mathbf{t})$	**Find elements connected in the given way (relation seeking)** Find the graph element(s) to which graph element $\mathbf{g_1}$ is connected in the given way at time $\mathbf{t}$: $?\ g_2 : (\mathbf{g_1}, \mathbf{t})\ \Lambda\ (g_2, \mathbf{t})$	**Find elements connected in the given way (relation seeking)** Find graph objects which are connected in the given way at the given time $?g_1, g_2,: (g_1, \mathbf{t}) \Lambda\ (g_2, \mathbf{t})$
	Neither time point specified	**Hybrid** Find the time points at which two given graph objects were connected in the given way $?\ t :(\mathbf{g_1}, t) \Lambda\ (\mathbf{g_2}, t)$	**Find elements connected in the given way (relation seeking)** Find the graph element(s) to which graph element $\mathbf{g_1}$ is connected and the time(s) at which the connection(s) occur $?\ g_2, t : (\mathbf{g_1}, t)\ \Lambda\ (g_2, t)$	**Find elements connected in the given way (relation seeking)** Find graph objects (and their associated time points) at any time that are connected in the given way $?\ g_1, g_2, t :(g_1, t) \Lambda\ (g_2, t)$

Figure 49 Summary of elementary comparison and relation seeking graph structural tasks concerning graph elements, including variations involving specified and unspecified time points. For an explanation of the formal notation shown here, please see Appendix A.

5.4.1.5 Summary

The task matrices were developed as a systematic way to capture the many variations in tasks specified in the Andrienko framework, for the temporal graph case, which involves multiple referrers. The matrices relate to the original Andrienko framework as follows:

- Each matrix captures a general task type: lookup, comparison, relation seeking.
- Elementary and the three varieties of synoptic tasks are distinguished by the quadrants.
- Direct and inverse tasks are captured based on the extent to which the referential components are specified (i.e. for direct tasks, both time and graph are specified, and appear in the top left of each quadrant).

- Variations in the number of additional constraints, and whether the same or different time/graph referrers are involved in a task appear in the rows and columns of the matrix.
- Comparison/relation seeking with a specified component naturally emerges where all elements of one of the lookup tasks are specified (these cases are noted in the task matrices).

The vast majority of comparison and relation seeking tasks can be formulated to involve the same or two different attributes. Only those involving the same time and graph components cannot; this is noted in the matrices.

Figure 50 combines the task matrix structures of

Figure 46, Figure 47, and

Figure 48 with the task hierarchy to illustrate where the task variations are situated.

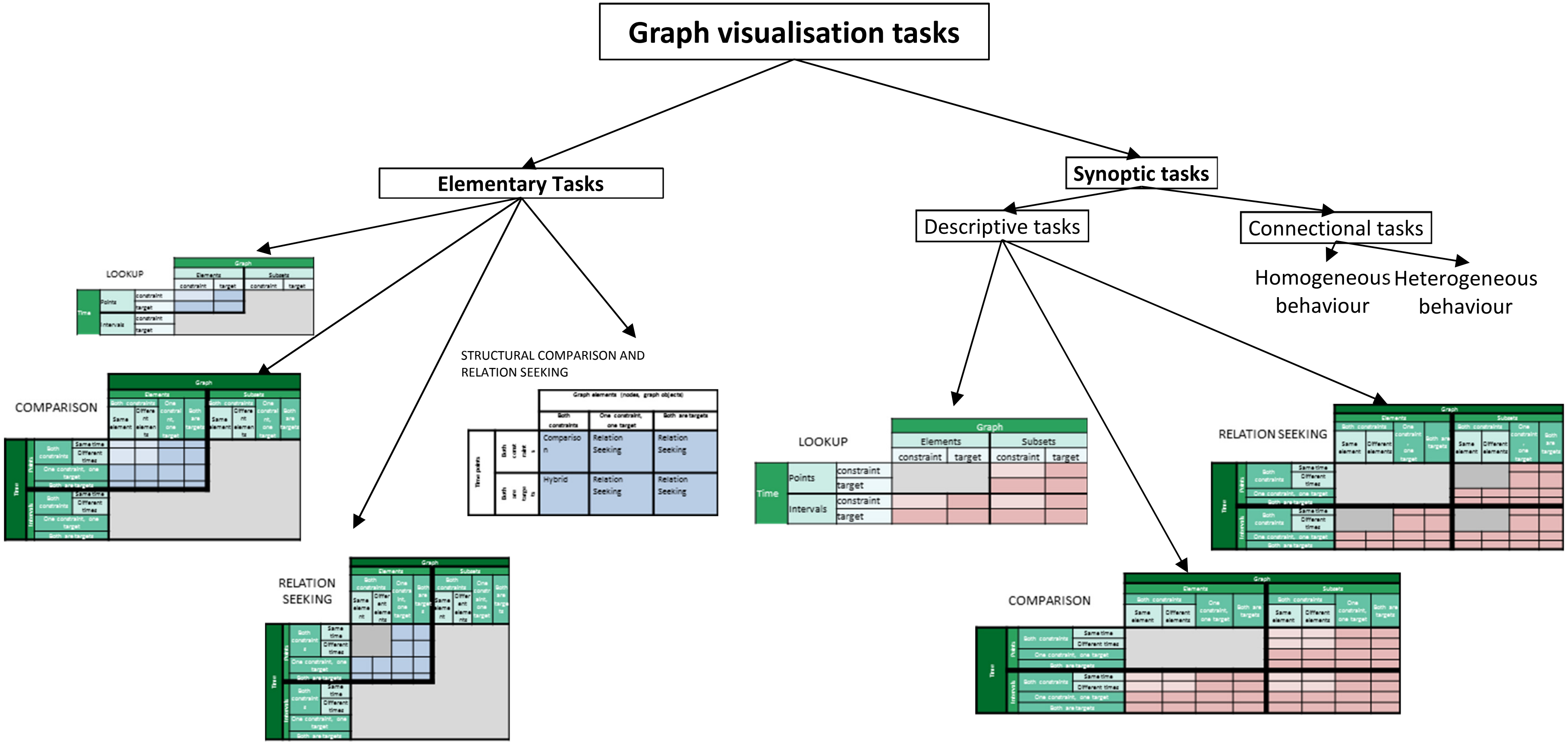


Figure 50 Task hierarchy (showing task types) combined with the task matrices (showing task variation). Note that the three task matrices are repeated for the set of synoptic descriptive structural tasks, which involves structural patterns, rather than attribute-based behaviours.

5.5 A ‘Slice and Dice’ approach to task classification

Andrienko and Andrienko’s work seeks to help designers of visual analysis tools. They identify both the tasks necessary to perform exploratory analysis, and the different types of visual tools available. However, they are unable to map the tools to the techniques which they support, and instead develop a set of general principles for selecting and designing exploratory tools. The reason that they do this is because of the difficulty inherent in mapping tasks to techniques in their general framework, which is applicable to all data types:

“…*The fundamental reason is that the tasks arising in data exploration are too specific (they are always formulated in the terms of data components), whereas the task categories that we identified are too generic. It is impossible to link each specific task to the appropriate tool(s) because the specific tasks are countless. Linking the tools to the generic task categories is also problematic, but for a different reason: the categories are so generic that no tool can perform all tasks belonging to the same category.”*

(Andrienko and Anrienko, [5], pp463-465)

However, they go on to describe the cases in which it *could* be possible to map tasks to techniques – for a specific dataset or a class of data sets.

In this work, a very specific class of data sets is considered: those involving temporal graph data. Individual data sets are included in this class purely by virtue of their data structure, that of having a temporal and graph referential component. While the type and number of attributes which individual data sets may contain may vary, the limitation on the referential component is sufficient to restrict the set of possible tasks. Moreover, through use of the task matrices, a full range of task permutations can be systematically specified. However, as noted earlier, these task matrices produce a very large set of tasks. In this section, useful ways in which these tasks can be classified are now considered. Part of the difficulty of classifying tasks is that each task can be categorised in more than one way; using the matrix structures as a guide, a ‘slice and dice’ approach can be used to group together tasks which will potentially require similar visual techniques for their support.

5.5.1 Temporal graph task categories

The first useful distinction relates to the classification of tasks by the referential components that participate, which results in the quadrants of each task matrix. This distinction is fundamental when selecting the most appropriate visual approach, as it allows us to consider the different research areas to which tasks relate, and therefore, to which areas we should look for appropriate visual tools and techniques:

- Q1: general visual techniques
- Q2: static graph visualisation
- Q3: temporal visualisation
- Q4: temporal graph visualisation

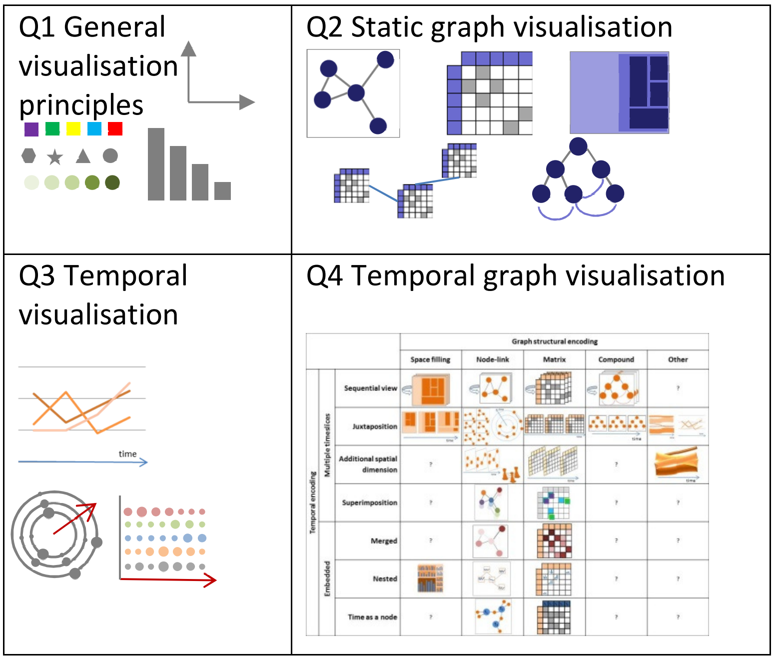


Figure 51 Research areas associated with data items by quadrant

This distinction highlights a key point: while Q4 is specifically related to temporal graph visualisation, every task in the matrix is a potential task when exploring temporal graph data; therefore visual techniques from *all* of these research areas may be involved when exploring temporal graph data.

As outlined above, Andrienko and Andrienko note that grouping by task categories (lookup, comparison, relation seeking) does not provide a useful basis on which to map techniques, as the tasks that fall into these categories are too generic for a single tool to support all of them. However, if we consider these tasks in each of the quadrants, we narrow down to a more useful classification, based on which we can begin to consider appropriate techniques. For example, lookup (usually) requires that we be able to see the data items for which we are looking: Figure 51 illustrates the idea that appropriate encodings for representing the different categories of data items will be found in different research areas. When comparing data items and finding relations between them, general visualisation principles to support comparison (such as alignment) are applicable to Q1. Gleicher et al.'s [102] three basic possibilities for visual comparison - juxtaposition (placing representations side by side), superposition (overlaying representations in the same display space) and explicit encoding (where the relationship between the two items is calculated and explicitly represented) - are applicable to the remaining quadrants. In addition, Q2 can draw on the large body of literature in graph comparison, using techniques such as animation, visual links, colour coding, and brushing and linking [103]. Comparison of data items in Q4 is not well documented in the literature, and there may be issues regarding the effectiveness of combinations of comparison techniques and the techniques used to show these data items (for example: would comparing sequential (animated) views side by side be an effective way to compare structural change over time in two different graphs?). These ideas are discussed further in Chapter 8.

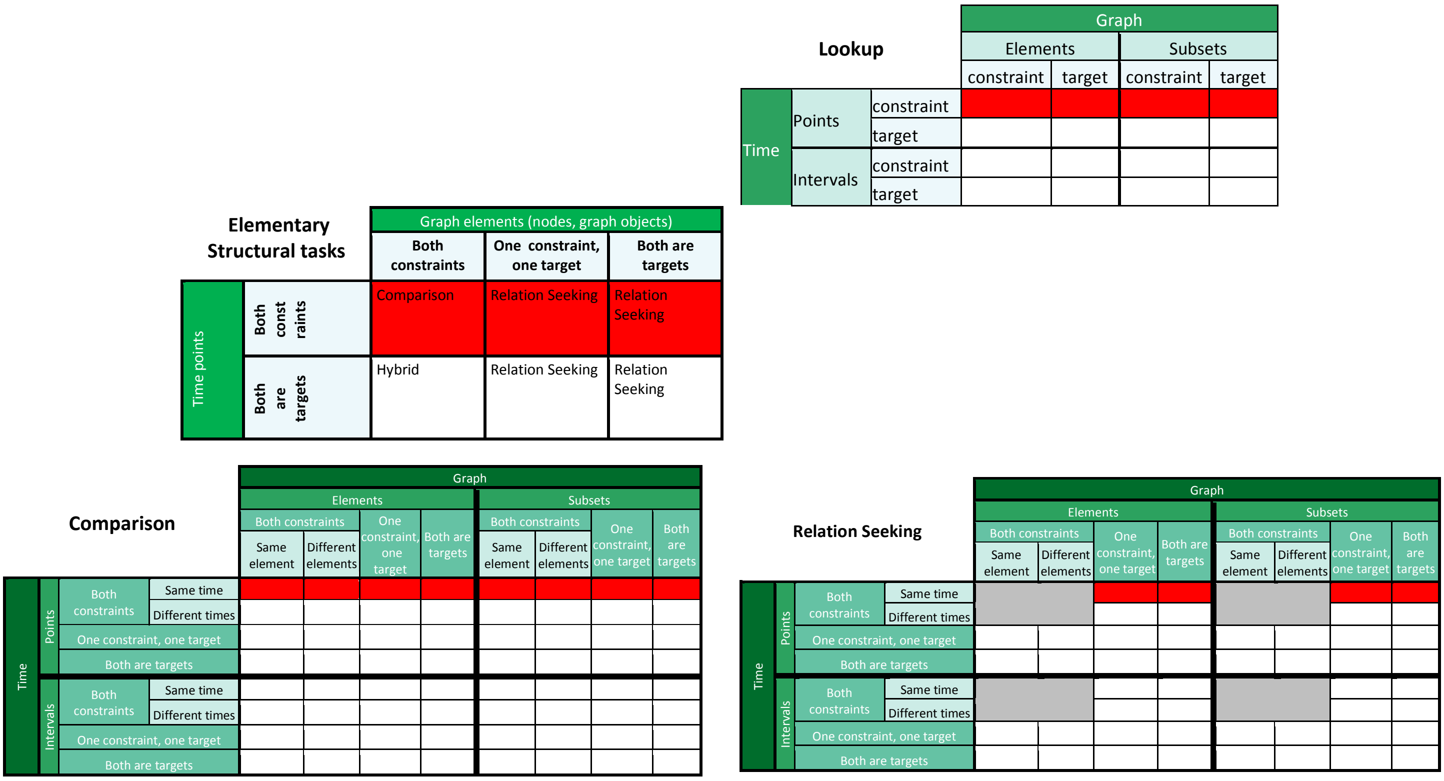


Figure 52 Task matrices showing static graph tasks (highlighted in red)

The rows and columns of the task matrices (across task types) are another useful way in which we can categorise the tasks. The set of static graph tasks appear in the rows of the matrices involving a single time point as a constraint (Figure 52). In graph comparison, while lookup, comparison, and relation seeking tasks are all relevant, only elementary tasks and those involving graph structure (i.e. not trends over time) are applicable. These can be clearly identified as the tasks which fall under Q1 and Q2 in the matrices.

The distinctions relating to same/different graph/time components and additional constraints are neatly captured in the rows and columns of the matrices. For example, comparison and relation seeking tasks involving two different time points or intervals can be easily distinguished (Figure 53). This is an important distinction as comparison between time points/intervals requires showing two graphs (or two graphs over time), as opposed to the single graph, or graph over time, representation required when the task involves the same time component.

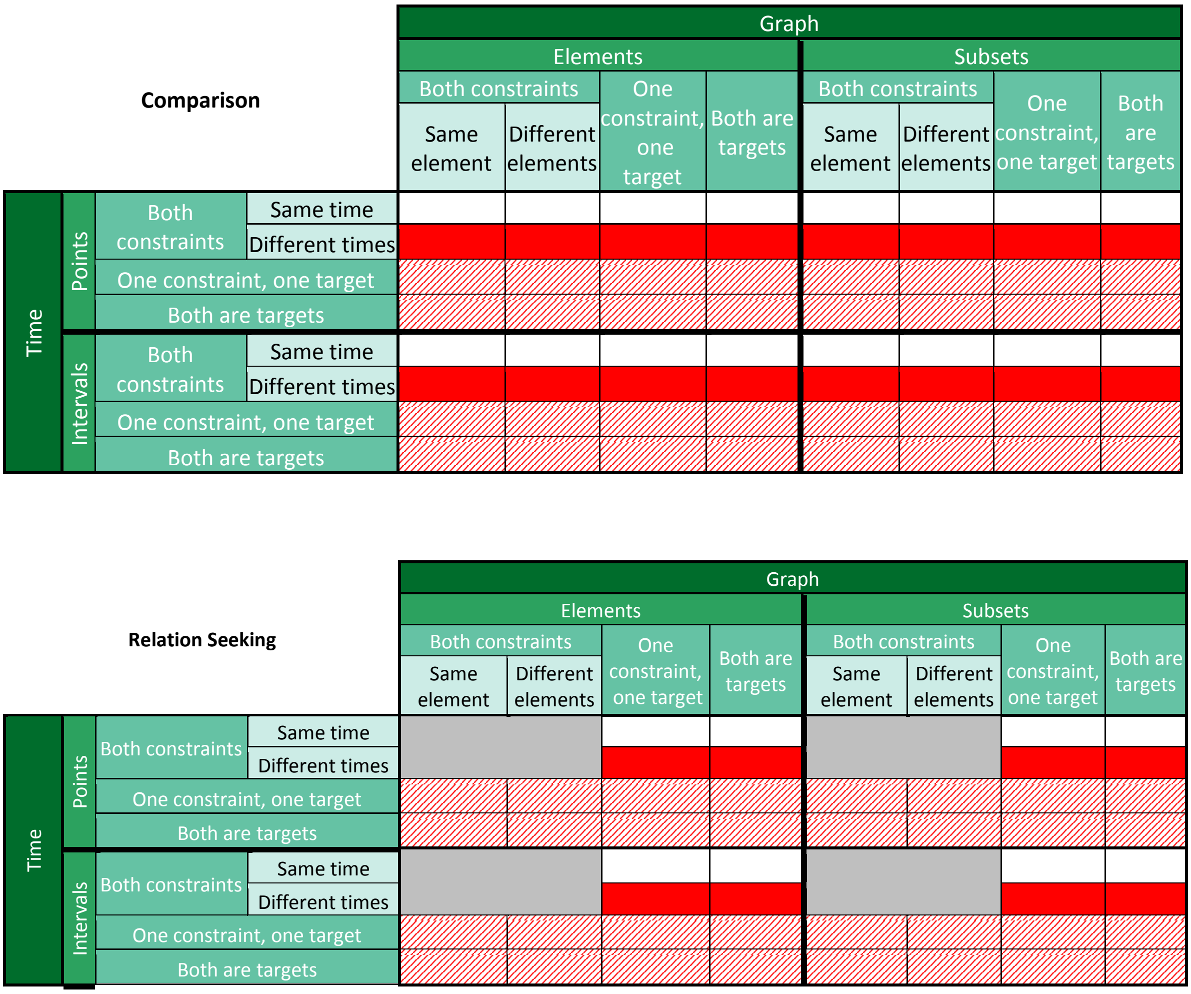


Figure 53 Comparison and relation seeking tasks (potentially) involving different time components e.g. where the time component is not specified (a target), an additional constraint can be added to the task specifying that the unknown time component is a different time point or interval. (Solid red shading indicates tasks involving a known different time component; cross hatching indicating tasks where time components are *potentially* different).

A further distinction which can be seen in the matrices is between what can be termed ‘evolutionary’ and ‘contextual’ tasks. Evolution - the notion of change in some object over time, be it the graph structure or its substructures, the attribute value of an individual node, or the distribution of attribute values over the graph - is often of interest when investigating temporal graph data, and this is reflected in the predominance of evolutionary tasks found in the literature. The task matrix structure easily distinguishes - but does not limit us to consideration of - evolutionary tasks, which involve a combination of the same graph element or graph subset at different time points or over different intervals[28] (evolutionary tasks are highlighted in red in **Figure 54**). Contextual tasks consider an object in the context of other objects, which may be at the same or different times. The range of such tasks identified in the taxonomy reminds us not to neglect the questions which enable us to situate our findings and bring perspective to our observations, perhaps in turn bringing deeper meaning to our study of evolutionary changes.

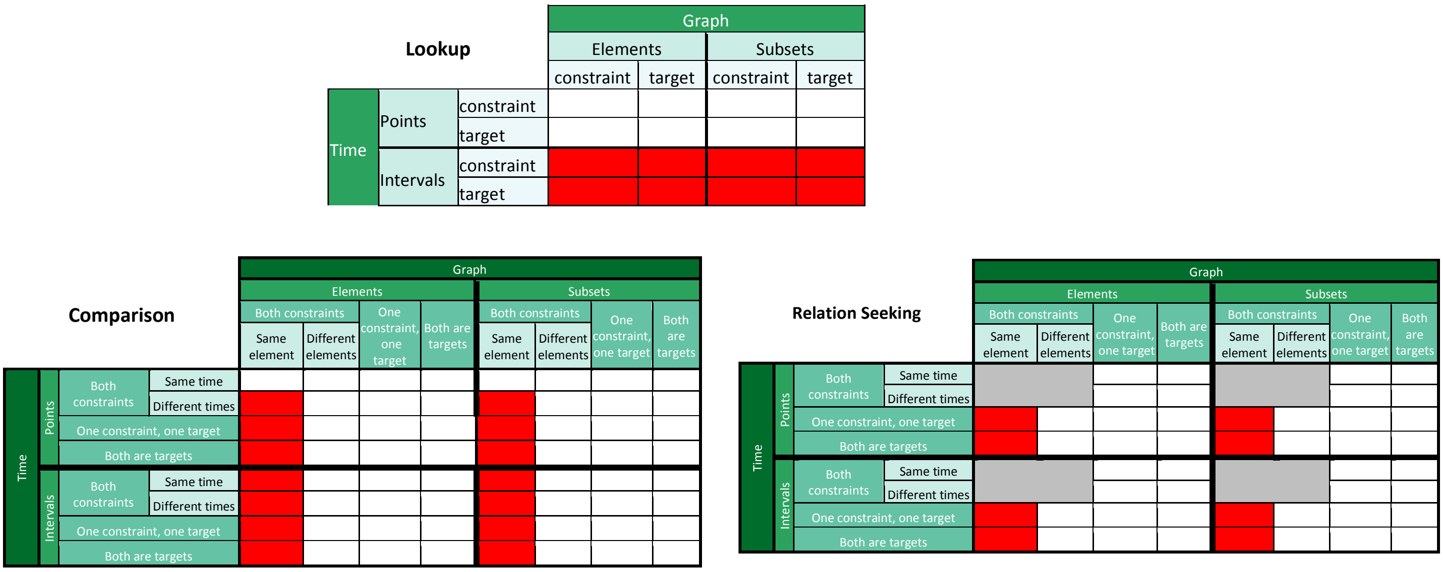


Figure 54 Evolutionary tasks involve the same graph element at different times

An important distinction when selecting a visual representation is the notion of task search space. Search space is dependent upon which data components are specified. Farrugia and Quigley [78] considered temporal search space, distinguishing between local (focussing on a specified time period), and global (searching across the entire time period).

This notion can be extended to consider the graph search space, giving four variations of task search space:

- no search (both time and graph components are specified)

[28] This should not be confused in general with the tasks of Q4 which involve graph structures over time. For example, comparing two different graph structures over the same time period would be a contextual task, whereas comparing the same graph structure over two different time periods is an evolutionary task; however, both tasks can be found in Quadrant 4 of the comparison matrix.

- graph search (time is specified but graph components are not - requires searching the entire graph)
- temporal search (the graph component is specified but the temporal component is not - requires searching the entire time period)
- graph and temporal search (neither component is specified - requires searching the whole graph at all time points).

Search space is related to the inverse/direct distinction of the framework, but is independent of the elementary/synoptic distinction. Even in elementary tasks involving a single element at a single time point, where both components are unspecified, the search space extends to the entire graph over the entire time period. The task search space is clearly identified in the task matrices by the columns and rows indicating the specified and unspecified referential components, with the widest task search space is to be found at the bottom right of each quadrant.

In addition to tasks in the matrices, additional tasks can be formulated to involve comparison with specified attribute or referential components. In this case we may need some additional way to visually represent the specified pattern.

The majority of tasks in the comparison and relation seeking matrices can also be formulated for tasks involving two different attributes. Again, such tasks may warrant different visual approaches (i.e. when making comparison between two different attributes as opposed to a single attribute).

One final classification regards the distinction between structural and attribute-based tasks. In the extended taxonomy an additional category of purely structural graph tasks was introduced. While this category sits separately from the attribute-based tasks, the picture is more complex than indicated by the existing structural vs attribute distinction made in the task literature: for example, Lee et al. [40] separate their topology and attribute based graph tasks into distinct categories; Ahn et al.'s [41] 'property' dimension distinguishes structural attributes and domain properties. However, partitioning tasks into those purely involving structure, and those purely involving attributes, is not helpful when considering visual approaches, as it ignores the middle ground in which a large proportion of graph based tasks reside i.e. tasks involving consideration of attribute values in the context of graph structure. There are therefore three possible categories of tasks:

- **Structural (no attributes involved)**: these tasks solely consider the structure of the graph, without reference to attributes. These are the structural tasks identified in the extended taxonomy. Visualisations supporting such tasks focus on representing the graph structure alone.
- **Attribute-based in a structural context**: these consider patterns of attributes over the graph structure and the position in the graph of the occurrence of attribute values. These tasks are captured in the attribute based tasks of the framework using the Andrienko behaviours. Visualisation approaches supporting these tasks require representation of the attribute values in the graph structural context.
- **Attribute based**: these consider attribute values in isolation from the graph structure. We may only be interested in attribute values associated with a graph in their temporal context e.g. the temporal trend in an attribute value for an element or set of elements (A2). We may also be interested in the frequency distribution (rather than the structural distribution) of the attribute values of all graph elements at a given time point, and how this distribution changes over time. Visualisation approaches which do not involve the graph structure e.g. [104] are appropriate in this case. This category also covers changes in structural metrics, which in themselves capture the structure of the graph, hence do not require an explicit structural representation when visualising them.

5.6 Connection discovery in temporal graphs

So far, only the descriptive tasks for temporal graphs have been considered. Examples of the three variations of relational behaviours involved in connection discovery tasks in temporal graphs are discussed in this section, and how they are applicable in the temporal graph case. The relational behaviours in the Andrienko framework only involve the relationships between attribute-based behaviours. As the graph referrer has non-fixed linking relations between its elements, two additional possibilities can be considered for the connection discovery tasks in the graph case: the relational behaviours between graph structures and the connections between graph structure and attribute value.

5.6.1 Connection discovery in the Andrienko framework

As noted in Section 3.4.3.4, connection discovery tasks involve relational behaviours. Relational behaviours essentially involve the relation or connection between two behaviours: the “linkage” patterns describing them are correlation, dependency or influence, or structural connection. Andrienko note that this idea in some ways overlaps with the descriptive behaviour comparison task, as it involves the relation between two behaviours. However they stress that relational behaviours go further than just describing the similarities and differences between behaviours, as they involve determining the connections between behaviours. They illustrate this with a scatterplot example: comparing the distributions of two attribute values does not allow us to determine whether a correlation exists, but plotting them against one another in a scatterplot can indicate this relationship. Such a representation does not show the individual behaviour of either attribute, but reveals how they behave with respect to one another ([5], pp.126-127), hence the term “mutual” or relational behaviour.

Andrienko consider three scenarios in which relational behaviours might be sought. First is the case discussed above, involving two (or more) different attributes defined on the same reference set. The second case widens this definition to involve different reference sets, however, it is noted that it is highly unlikely that the two reference sets would be completely unrelated. An example given is that of the connection between the spatial position of rings in a tree trunk and climate over time. Both of these cases involve the connections between two or more phenomenon and are therefore termed heterogeneous behaviours.

The last case considered is that involving internal connections within a single phenomenon, termed homogenous behaviours. Andrienko give examples of investigating the relationship between the pre- and post-natal development of a baby, or frequent associations occurring in a data set, such as hot summers being followed by cold winters ([5], p130). The formal notation relating to connection discovery behaviours is included in Appendix A.

5.6.2 Additional possibilities for connection discovery in the graph case

Let us now consider relational behaviours in the graph case. Recall that the graph referrer of the extended data model is unlike the other referrer types, as the relations between its elements are not fixed. There are therefore some additional possibilities for connection discovery in the graph case.

In the case of descriptive behaviours, the distribution of attribute values over the graph is determined by the relations between referrers; a behaviour characterisation task therefore involves describing this distribution. However, we may want to go further and investigate, for example, whether a node’s position in the network in some way *influences* its attribute value. This sort of question is the subject of egocentric social network analysis, which uses the structure of a node’s ego network to predict, for example, an individual’s health or economic status [105]. In the temporal graph case, we may also be interested in the effect of network structure in determining the distributions of attribute values. For example, in sociocentric social network analysis the relationship between network structure and attribute values is investigated to determine how network structure affects concentration of power, access to new ideas, and spread of disease etc. [105] e.g. Christakis and Fowler’s [106] investigation into the influence of network structure on obesity. Additionally, we may also be interested in the effect that attribute values have on the network’s structure e.g. do certain levels or patterns of attribute values precede particular structural changes?

A second possibility in the graph case is to consider the connections between structural behaviours. Here we may be interested in the relationship between the structures of different parts of the graph, or in the temporal case, how structures at one point in time influence structures at another. For example, in social network analysis a number of theories surround tie formation. Yi et al. [70] discuss examples of these including preferential attachment, accumulative advantage (actors with many ties gain more ties), homophily (the theory that those with similar traits connect to one another), follow-the-trend (i.e. the dominant choices of others), and multiconnectivity (a pursuit for diversity and multiplexity). In all cases, we would look at how structural patterns in the graph at one point in time influence the structural patterns at another.

The additional possibilities for the graph case so far can all be considered to be variations of homogenous behaviours. Let us now consider some examples of heterogeneous behaviours in the graph case. Where two different attributes are involved, we could consider the relationship between attribute values without reference to the graph structure e.g. '*does publication count depend on department*?' or we could additionally consider the graph structure e.g. whether there is some connection between the two distributions over the graph.

Where two different reference sets are involved, we could consider for example, how the attribute values in the graph are influenced by outside events over time. This may be of particular interest, where some form of external intervention in the network is under observation, such as vaccination in a public health network.

We can also think of examples involving the connections between two different graphs. This behaviour may be of interest where we are investigating two different networks which are related in some way, for example, co-authorship networks from different domains, or the energy grid and a computer network. Moreover, we might not only be interested in attribute based behaviours, but also the relation between structural patterns. For example, like Gloor and Zhao [75], we might be interested in the relationship between networks constructed to reflect different communication mediums e.g. face-face, telephone, email. In this case we might also wish to find some correlation between the structural patterns of the network itself, for example, we can imagine that the times at which the email network is densely connected, the face-to-face network may be less so.

5.6.3 Summary

While descriptive tasks aim to describe individual behaviours, the connection discovery tasks of the Andrienko framework concern relational behaviours, which can be described by linkage patterns such as correlation, dependency, and structural connection. The Andrienko framework considers connectional tasks according to whether they involve:

- Heterogeneous behaviours: relational behaviours involving two (or more) different attributes over either
  - the same reference set or
  - two different parts of the same reference set
- Homogeneous behaviours: relational behaviour involving the same attributes of different reference subsets

While all of the tasks in the Andrienko framework concern relational behaviours arising from the interaction of attribute-based behaviours, for temporal graphs, two additional possibilities can be considered:

(1) relational behaviours arising from the interaction of graph structural behaviours, and

(2) relational behaviours between graph structure and attribute values.

Note that the task types of the connectional tasks are the same as those of the descriptive tasks (lookup, comparison, relation seeking), but they involve relational behaviours and linking patterns.

5.7 **Summary**

This chapter has demonstrated the application of the extended data model and task framework outlined in Chapter 4 to the specific case of temporal graph data.

By considering the possible combinations of time and graph (referential) components (and associated data items) that may participate in temporal graph tasks, an additional data based distinction extends the Andrienko notion of elementary and synoptic tasks. This results in four fundamentally different task categories ("quadrants") in the case of temporal graph data, any or all of which may be required during analysis. These four task categories can be combined with the task types (distinguished based on the data items which participate as targets and constraints) to produce a wide range of tasks. In addition, the Andrienko framework specifies a number of sub variations within the task types. In order to systematically specify the possible task variations, a set of task matrices were developed, which capture the possible combinations of specified and unspecified referential components, and additional variations in whether the graph and time components are the same or different. In addition to allowing systematic specification of task permutations, the matrices allow for a "slice and dice" approach to be taken to task classification. All of the tasks can be classified along multiple dimensions, and therefore fall into multiple categories. The dimensions upon which tasks can be usefully categorised were summarised in Section 5.5. These categories are used in Chapter 8 as the basis upon which to map tools to tasks. The connection discovery tasks which may be relevant when exploring temporal graph data were also considered, including relational behaviours between graph structures, and between graph structure and attribute values.

In the next chapter, the developed temporal graph task framework is evaluated.

Chapter 6 Evaluating the task classification

Having used a formal approach to generate a task design space for temporal graphs in Chapter 4 and Chapter 5, this chapter focusses on its evaluation. Existing strategies for evaluating task classifications in the visualisation literature are first considered. Drawing on the evaluation practices identified in the literature, the task classification is firstly evaluated with respect to the extant temporal graph task frameworks. Secondly, an empirical study to evaluate the utility of the framework's use in the design process was carried out and is reported on.

6.1 Existing evaluation practices

While practices for evaluating visualisation systems and techniques have become a topic of increasing interest to the visualisation community (e.g. [6], [9], [107], [108]), very little attention has so far been given to the evaluation of formal models utilised by the community, such as task classifications. While we would expect a publication demonstrating a new visualisation technique or system to include some form of evaluation with respect to its utility, performance, and limitations, this does not appear to be the case when newly developed classifications are reported. Indeed, several papers presenting task classifications contain no evaluative component at all. The lack of consideration given to evaluating classifications is surprising, given that measuring the effectiveness of task classifications has been recognised as a difficult problem [13] and the benefits of evaluating classifications parallel those of evaluating visualisation systems, including: identifying areas for improvement resulting in better classifications; convincing potential adopters of the validity and utility of the approach (particularly important for more complex classifications which may require significant effort to adopt); and helping adopters select between competing classifications.

Given the lack of formalised guidance on the evaluation of task classifications, the literature was reviewed in order to establish: (1) the aspects of task classifications that have been evaluated, and (2) the methods employed in evaluating these aspects, particularly with regard to 'downstream' evaluation practices, where a final, fully developed classification is evaluated. Considering the threats to validity and appropriate approaches to validation at each stage of the construction process also forms an important part of the evaluation process, and was considered in Section 3.1. Rind et al.'s list of 31 abstract task categorisations was again used as the basis for the literature review (see Section 3.1 for the publications included in the review, also summarised in Table 1). Literature relating to the visualisation design process and evaluation practices was also drawn on where appropriate.

6.1.1 Overview

In reviewing the literature it was found that evaluation of task classifications is lacking. The definition of evaluation used in the review was broad, in that any discussion regarding the limitations of the classification or its relation to other works was considered to be a form of evaluation. Yet 9 of the 26 papers reviewed offered no explicit evaluation. In many cases, where discussions were included, these reflections were brief and unsystematic (they were perhaps not intended to serve the purpose of evaluation). The vast majority of evaluations in the review were discussion based (either discussions of limitations, discursive comparisons with extant classifications, or discussion in relation to some predefined evaluative criteria (see Section 6.1.3)). Only 8 papers used an empirical approach, such as use of the classification in the design or evaluation of systems, or testing it with domain experts. While discussions are a valid form of assessment, the brevity and lack of rigour in some indicate that this topic could benefit from more attention.

In the following discussion evaluation strategies are distinguished according to *what* they seek to evaluate: the method of construction, properties of the classification (descriptive powers, comprehensiveness, real world nature of tasks, syncretism, usability), and use of the classification (e.g. in the design or evaluation processes). These latter two strategies are of course not entirely distinct, as through evaluating the use of the classification, authors often seek to evaluate the fundamental properties of the classification. Finally, adoption rates are discussed as an additional method of evaluation. While reviewing the literature a distinction in evaluation practices was also noted between classifications which seek to improve upon extant classifications (and thus the need to evaluate in relation to other works in order to demonstrate some comparative advantage) and those which seek to unify extant works (which need to demonstrate that they have the ability to capture all aspects of extant classifications). The discussion is summarised for reference in Table 22.

6.1.2 Evaluation of construction method

While the threats to validity and potential validation approaches at each stage of classification construction were identified in Section 3.1, these were not widely discussed in the literature reviewed. However, consideration of these methods can play an important part in validating the classification. Several papers did reflect on the construction method to some extent when discussing the limitations and advantages of their work. Roth et al. [62] suggest that the empirically-derived nature of their framework makes it ecologically valid, and therefore offers advantages over other works. Amar et al. [36] reflect on the limitations of using student questions as the basis of their classification, while Brehemer et al. [53] acknowledge the limitations of using interviews to gather task sequences, noting that their classification may be incomplete due to sampling or observer bias. Wehrend and Lewis [8] consider the rigour with which their categories were selected, and whether an abstract mathematical approach would provide a “cleaner” analysis.

6.1.3 Evaluation of classification properties

Let us first consider the fundamental properties of a classification that were found to be subject to evaluation. Two papers explicitly validate their classification in relation to a set of pre-identified criteria. Yi et al. [13] discuss their classification with respect to Beaudouin-Lafon’s [109] dimensions to evaluate interaction models - *descriptive power*, *evaluative power*, and *generative powers* (which relates to use in the design process). Sedig and Parsons’ [47] consider their classification in relation to four characteristics: *syncretic* (its ability to unify previously disconnected ideas), *general* (in its level of abstraction and applicability), *comprehensive* (in terms of coverage), and *generative* (in relation to use in the design process and guiding future research).

Ahn et al. [41] design an empirical evaluation in which they seek to validate *comprehensiveness* (task coverage); *ease of use*; *precision* (descriptive power); *use in task organisation and clarification*; *use in task discovery* (i.e. as a generative method during the domain problem characterisation stage of the design process).

As mentioned above, evaluation with respect to *use* of the classification (e.g. in the design and evaluation processes) is considered separately from the evaluation of properties in this discussion. Beyond these three papers, *descriptive powers* and *comprehensiveness* were also the main properties evaluated more widely in the literature. Verifying the *existence of tasks in the "real world"* and usability of the classification was also found to be of interest. The methods for evaluating each of these properties are now considered.

6.1.3.1 Evaluating descriptive powers

The fundamental purpose of a classification system is to use a common language to be able to describe the full range of tasks in a consistent way. A common method to evaluate this ability is to use the classification to describe a set of known tasks and check that they can all be adequately captured. Examples include using the classification to describe: the tasks which can be supported by an existing system [46], [44]; those common to a specific domain [45] or identified by domain experts [41]; or sequences of tasks carried out by people using a visualisation system [7]. Brehmer and Munzner [67] go one step further by explicitly comparing their classification's ability to describe a sequence of tasks supported by an extant system, with task descriptions generated by other classifications. Thus they are also able to demonstrate how their classification overcomes the shortcomings of others in terms of its ability to describe.

6.1.3.2 Evaluating comprehensiveness

A classification's ability to capture all possible tasks is in many respects related to its descriptive powers. However, to evidence that a classification is *complete* is rather difficult; as per the problem of induction, we are always just one task away from finding a case which our classification cannot cover. It is particularly difficult to demonstrate when the taxonomy is intended to be useful across multiple domains, with a wide range of possible tasks. Additionally, classifications may not be able to capture tasks specified at multiple levels of composition (i.e. high or low level tasks). Discussions relating to limitations often seek to demarcate the limits of the claimed comprehensiveness of a framework e.g. with regards to tasks omitted or those that lie out with the intended scope of the classification e.g. [45], [36], [53]. While not showing completeness, demonstrating that the developed classification is able to cover more tasks than extant frameworks is one form of validation. Similar to the evaluation strategy outlined with regard to descriptive powers, mapping a large set of tasks e.g. from problems identified in the literature to the task categories [8], may also go some way to demonstrate task coverage. Finally, where formal modelling approaches have been employed in the classification's construction, a formal proof can be used to demonstrate comprehensiveness of the classification, at least with respect to the chosen model [5].

6.1.3.3 Evaluating the "real world" nature of tasks

As discussed in Section 3.1.4.1 classifications developed using formal modelling techniques e.g. [5], [49], or design spaces, where all possible permutations of tasks are generated by combining dimensions, (e.g. [37], [41]), leave open the question of whether the generated categories are merely constructs of the formal process or are in fact representative of 'real world' tasks. Validation of such frameworks may therefore involve establishing that the tasks are indeed 'real world' tasks. This type of validation is usually dealt with in the literature by providing illustrative concrete examples for each possible category of abstract task. The most comprehensive example is probably Andrienko [5] who include several data scenarios from different domains which they use to provide examples to illustrate most of the possible iterations of tasks generated by their modelling approach. Sedig and Parsons [47] offer examples of existing tools which implement each of their patterns in order to evidence the existence and necessity of each pattern in the real world. However, few frameworks offer an example task for every possible permutation of their model. Evaluating the 'real world' nature of tasks can therefore prove tricky. Simply because we cannot readily think of a concrete example of a task category, it does not mean that it is not a real task, albeit perhaps exclusive to a particular domain or niche analysis scenario; Schulz et al. [37] note that “*what looks like an inconsistency in the design space may actually be just a very creative and unusual combination of design choices*”. For the more extensive frameworks, examples may need to be drawn from multiple domains to cover all possible tasks, which may require input from multiple domain experts.

6.1.3.4 Evaluating syncretism

For classifications which seek to unite extant classifications, it is important to show that extant categories can be subsumed under the proposed system. Often categories are compared through discursive methods e.g. [44], sometimes highlighting the advantages of the proposed classification e.g. [64]. A more rigorous approach is to explicitly map the categories of extant frameworks to those of the proposed framework, as done by Brehmer and Munzner [67]. The resulting mapping not only shows where the categories sit, but also reveals which categories are under- and over- represented in previous works. However, comparing categories between classifications may not be straightforward: Schulz et al. [37] note difficulties including that categories which have been separated out in one work may be mixed in another, or authors may have fundamentally different ideas about what a task is (such as combining objectives and actions).

6.1.3.5 Evaluating usability

Ahn et al. [41] was the only paper reviewed which assesses the use of the classification by someone other than the classification developer, allowing them to evaluate its usability. They do so as part of their empirical study which involves interviews with domain experts. Note that most classifications are intended to be used by visualisation researchers (as opposed to domain experts), in which case evaluation with other visualisation researchers in terms of their usability would be more appropriate.

6.1.4 Evaluation of usage

While a wide range of usage scenarios for task classifications are identified in the literature (see Section 2.3), and the envisaged uses of the developed classification are often outlined in detail (e.g. [53], [67]), only a few papers evaluate their classifications directly with respect to their intended usage. Evaluating a task classification by employing it in an intended usage scenario provides us with information relating the utility of a classification for its intended purpose, and may also provide opportunities to indirectly evaluate the properties of the classification, as described in Section 6.1.3.

Use in the design process: Two papers include empirical evaluations of use in domain characterisation and abstraction. Ahn et al. [41] use interviews to explore the use of their classification as a generative method when establishing tasks of interest to domain experts, and its ability to help them organise, describe and clarify their tasks. Schulz et al. [45] report on a use case with domain experts. They use their classification to characterise and organise known tasks, establishing the most common and important tasks. Having also characterised existing tools according to the tasks that they support, they are also able to use their classification at the encoding/interaction technique design stage of the design process, selecting tools which are able to support the identified tasks.

Other studies report more generally on the use of their classification in guiding the design process. Amar and Stasko [46] demonstrate the use of their framework in a hypothetical design scenario, in order to illustrate its use as a "*systematic basis for thinking about and identifying issues in the data set*." However, they do not develop (and therefore do not evaluate) the resultant system. Wehrend and Lewis [8] used their catalogue to develop a visualisation, presumably using it to select appropriate techniques for inclusion (they do not give a detailed report regarding its use in the design process). They do not formally evaluate the resulting representation, but conclude that it "*appears to be an improvement over earlier representations designed in an ad hoc manner*". This highlights one difficulty with this kind of validation, in that it can be difficult to say to what extent the classification was useful in guiding the design. For example, had the design process proceeded without the use of the classification, would the resulting system have been any different, or in some way less good?

Two papers ([63] and [64]) - both concerned with analytical provenance - directly implement their classifications in the design of a system in order to track users' analytical processes. Gotz et al. [63] seek feedback from developers regarding the ease of implementing the model, and interview analysts who used the system with regard to how well it aligned with their mental models.

Use in evaluation: while many ways in which task classifications can be used in evaluations were identified in the literature (see Section 2.3.5; indeed, this is often cited as the primary motivation/purpose for their development), only one type of evaluation scenario was included in the body of work reviewed: using the classification to characterise extant systems, then comparing them according to task support [46], [45],[44],[65], [40]. Sacha et al. [44] also include a variation of this scenario, where they use their classification to assess a visual analytics system in terms of how it supports different aspects of their classification. This allows them to point to shortcomings and areas for improvement in the system's design.

6.1.5 Evaluation with respect to adoption

One final evaluation strategy identified in the literature is that of adoption rates: Heer and Schneiderman [48] suggest validation via "*community feedback, critique and refinement*". The adoption, evolution, and demise of task classifications 'in the wild', may provide significant information about their descriptive abilities, comprehensiveness, usefulness, and usability. Where limitations in task coverage are encountered, classifications are often adapted or extended (e.g. the extension of the Andrienko framework [5] to temporal data by Lammarsch et al. [49]), unified e.g. [45], [67], or new classifications are developed. Where task classifications are found not to be useful they are likely to be superseded. Even where a classification may offer better descriptive abilities or more comprehensive task coverage, in the busy world of visualisation research, classifications which are easy to understand and require little learning overhead may be more likely to succeed.

Aspect	Method
Construction method	Critique of method employed [62], [46], [53], [8]
Property: descriptive power	- Use classification to describe a known set of tasks: from existing systems [46], [44]; common to a specific domain [45]; identified by domain experts [41]; carried out by users of visualisation systems [7]; problems in the literature [8] - Compare with other classifications' descriptive powers [67]
Property: comprehensiveness	- Discussion of limitations [53], [45], [66] - Demonstrate able to cover more tasks than extant works - Describe a (large) known set of tasks (as per descriptive power) - Formal proof (formal modelling processes only) [5]
Property: real world nature of tasks	- Provide illustrative concrete examples [5], [47] from across multiple domains. - Input from domain experts
Property: syncretism	- Discussion [44], [64] - Map categories of extant classifications to proposed classification [67]
Property: usability	Assess use of classification by intended users; interviews [41], [63]
Usage: design process	- Empirical evaluation using the classification in the design process: as a generative method/task organisation [41]; in task organisation, tool selection [45] - Demonstrate use via hypothetical design scenario [46] - Report on results of using classification in design process [8] - Implement classification (for analytical provenance) [64], [63]; interviews to assess ease of implementing [63]
Usage: evaluation	Demonstrate use in evaluation process e.g. use of classification to characterise and compare task support in extant systems [46], [44], [45], [65], [40]; evaluate an individual system in terms if task support [44]
Adoption	Adoption rates as indicator of validity of classification.

Table 22 Aspects of task classifications which can be evaluated and associated evaluation strategies

6.2 Evaluating the developed task classification

As discussed in the previous section, a task classification can be evaluated with respect to four aspects: the construction process; the properties of the classification; the intended usage scenario; and adoption rates.

The limitations of the *construction method* used in this work were discussed in Section 3.2. This identified the lack of involvement of people as one of the main drawbacks of using a formal modelling process to construct a task classification, particularly with respect to task coverage and establishing the real-world nature of tasks. In order to validate the task classification developed in this thesis with respect to these *properties*, further work is required to establish:

(1) whether the model is sufficient with respect to task coverage

(2) whether the tasks are ‘real-world’ or constructs of the formal process

As there already exist a number of classifications of temporal graph tasks, it is also useful to show how this work compares with and improves upon those classifications. Further, as outlined in Section 6.1.4, in order to demonstrate the utility of a classification, it can be evaluated with respect to an *intended usage scenario*.

Taking into consideration these aspects requiring validation, and the previous discussion relating to evaluation practices, the following strategy to evaluating the classification is adopted:

Firstly, in Section 6.2.1, the developed task classification will be considered in relation to existing frameworks, particularly in terms of *comprehensiveness and descriptive powers*. Demonstrating that the developed classification is able to capture the tasks of extant frameworks, and also tasks which the extant frameworks are not able to capture, will provide further information about these aspects. Given that three of the four extant classifications were constructed using taxonomic methods (only one uses a formal approach), this may also offer us some further information as to the *real-world nature of tasks*. In addition, the descriptive abilities of each of the frameworks in terms of Rind et al.’s three dimensions (perspective, abstraction, composition) will also be discussed.

Secondly, an empirical study is presented, which uses the developed classification as a generative method at the task understanding and abstraction stages of the design process. This primarily demonstrates the *utility* of the classification in a usage scenario, but also offers further information relating to the *comprehensiveness of task coverage*, its *descriptive powers* and the *real-world nature of the tasks*. As noted in Section 3.1.2.1, task typologies and design spaces do not provide information relating to the most frequently occurring tasks. However, the study demonstrates how the classification can be used to organise tasks and help establish which types of tasks are frequently occurring for a specific application. The design of the study draws on the designs of the empirical evaluations carried out by Ahn et al. [41] and Schulz et al. [45], and is discussed in more detail in Section 6.2.2.

Finally, in Chapter 9, a case study is presented, the first part of which involves using the classification to evaluate an extant visualisation system's design in terms of its support for the task categories, in a manner similar to that described by Sacha et al. [44] (as discussed in Section 6.1.4). This allows for the identification of shortcomings of the system and areas for improvement in the system's design, and demonstrates the utility of the classification in a typical evaluation scenario.

6.2.1 Considering the task classification in relation to extant works

In this section, the task classification developed in this work is compared to the following four extant classifications (outlined in more detail in Section 2.4):

- Lee et al.'s (2006) task taxonomy for static graph visualisation
- Yi et al.'s (2010) tasks for temporal social network analysis
- Bach, Pietriga, and Fekete's (2013) task taxonomy for dynamic graphs
- Ahn et al.'s (2014) task taxonomy for network evolution analysis

Let us first briefly consider how comparable these taxonomies are. As noted previously, Rind et al. [43] classify task categorisations along three dimensions: *perspective* (either *objectives* - i.e. questions about the data, or *actions* - i.e. means by which the questions can be answered), *composition* (the level to which the task has been decomposed into smaller sub tasks, ranging from high to low) and *abstraction* (either *concrete* - i.e. expressed in domain terms – or *abstract* – expressed in generic, domain-independent language). Both Ahn et al.'s and Lee et al.'s taxonomies are included in Rind et al.'s survey and classification, as is the published version of the task classification presented in this work ([111]). The classifications given in Rind's survey are included in Table 23, to which classifications for Yi et al.'s and Bach et al.'s taxonomies have been added.

	perspective		composition			abstraction			
	why	how	HI	IN	LO	GE	DA	DO	TO
Yi et al. [70]	•		•	◦				•	
Ahn et al. [41]	•			•	◦		•		
Kerracher et al. [111]	•			◦	•		•		
Lee et al. [40]	•				•		•		
Bach et al. [110]	•				•		•		

Table 23 showing the classification of the graph/temporal graph task taxonomies according to Rind et al.'s framework. The three levels of composition are high (HI), intermediate (IN), and low (LO); the four levels of abstraction are generic (GE), data type (DA), domain (DO), and tool architecture (TO). Table is ordered to highlight similarities and differences [• primary class, ◦ partial match]

All of the classifications consider analytical tasks (a "why" perspective). We can see that Yi et al.'s tasks are composed at a higher level than those of the other frameworks, and are also domain specific. The remaining frameworks are all categorised as "data type" along the abstraction level, and their tasks are of low to intermediate composition.

The level of abstraction used by a task classification affects its descriptive powers and potential task coverage. While Yi et al.'s tasks are described using rather generic language, they explicitly intend to describe the tasks of social network analysis, and the classification is derived from an understanding of that domain. It is therefore likely that these tasks will be less able to cover the tasks of other domains than frameworks intended to be domain independent (for example, Yi et al. do not mention tasks involving paths, which would be of high importance e.g. in transportation networks).

In terms of composition, describing tasks at a lower level of composition may be particularly advantageous during the design process, especially during the domain problem characterisation and abstraction design stages [43]. This is similarly the case when constructing a mapping between tasks and the techniques which are able to support them, as per one of the aims of this thesis. However, Rind et al. [43] also suggest that low-level tasks may be too trivial when intended for use in stimuli for certain types of experiment, while high-level tasks can be too open-ended for quantitative analysis of time and errors; a suitable level of composition therefore needs to be found when choosing tasks for use in evaluation scenarios.

Let us now consider how the categories of the extant classifications relate to those of the classification developed in this thesis. Given the difficulties noted previously in comparing task classifications [37], this discussion will refer largely to the high level distinctions of task type and quadrants.

Yi et al.'s taxonomy specifies three main tasks when investigating change in social networks, based on the levels at which graphs can be analysed ([70] p1035):

- Task 1—Analysis of temporal changes at the global level.
- Task 2—Analysis of temporal changes at the subgroup level.
- Task 3—Analysis of temporal associations among nodal and dyad level attributes.

These tasks are captured in the quadrants: Q1 and Q3 (node/dyad level) and Q2 and Q4 (subgroup and global level). As outlined previously, Yi et al.'s tasks are specified at a higher level of composition than those of the classification developed in this thesis, therefore a specific mapping to the task types of the taxonomy is not possible.

<table>
<tr><th rowspan="2"></th><th colspan="2">Lookup</th><th colspan="2">Comparison</th><th rowspan="2">Relation Seeking</th></tr>
<tr><th>Direct</th><th>Inverse</th><th>Direct</th><th>Inverse</th></tr>
<tr><td>Q1</td><td colspan="2">✓</td><td colspan="2">✓</td><td>✓</td></tr>
<tr><td>Q2</td><td colspan="2">✓</td><td colspan="2">✓</td><td>✓</td></tr>
<tr><td>Q3</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td>Q4i</td><td>✓</td><td></td><td></td><td></td><td></td></tr>
<tr><td>Q4ii</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Table 24 General mapping of Lee et al.'s tasks to the high level categories of the classification developed in this thesis. Examples of direct and inverse tasks are included, as are examples of both relation seeking and comparison, although a distinction is not explicitly made. Note that most tasks are structural, as attributes are treated separately and only a limited number of examples of attribute based tasks are given.

As Lee et al.'s taxonomy is intended for static graphs, almost all of their tasks can be positioned in Q1 and Q2; only their high level task '*how has the graph changed over time*?' considers the temporal dimension. While their discussion of static tasks is comprehensive and offers many useful real-world examples, the taxonomy outlined in this thesis offers a systematic way to specify the possible permutations of these tasks. For example, for tasks involving attributes on nodes, the general description, *find the nodes having a specific attribute valu*e, does not consider the opposite, direct lookup task, *find the values of specific nodes.* Similarly, their topological tasks are generally phrased for relation seeking, rather than comparison; comparison is only briefly mentioned for the whole graph case, omitting the possibilities amongst individual nodes or edges. They also separate their topology and attribute based tasks into distinct categories. However, through the notion of behaviours, the framework in this thesis makes clear the important relationship between attribute values and graph structure, and includes tasks involving attribute distributions over the graph. Three of Lee et al.'s tasks do not fit into the task taxonomy described in this thesis, because they do not involve questions about the data: '*Follow path'* and '*revisit*' are visual tasks (actions, to use Rind et al.'s [43] terminology), while *'give a meaningful name to a group'* is not a question asked of the data – it relates to an individual's interpretation of the data. The general mapping of Lee et al.'s tasks to the high level categories of the classification developed in this thesis are shown in Table 24.

	Lookup		Comparison		Relation Seeking
	Direct	Inverse	Direct	Inverse	
Q1	✓	✓			
Q2	✓	✓			
Q3	✓	✓			
Q4i	✓	✓			
Q4ii					

Table 25 General mapping of Bach et al.'s tasks to the high level categories of the classification developed in this thesis. Note that these tasks involve graph structure only.

Bach et al.'s approach is most similar to the approach taken in developing the taxonomy presented in this thesis. Their '*when*' dimension is captured by the temporal referrer, while the '*where*' dimension is the equivalent of the graph referrer. The *'what'* dimension is similar to the characteristic component of the data model used in this thesis; note, however, that only structural behaviours are discussed in Bach et al.'s framework, with no mention of attributes. Their approach allows them to capture both direct and inverse lookup tasks. However, they do not consider relations and attributes as separate data items (these are incorporated into the ‘*what*’ and ‘*where’* dimensions). This makes their task framework less complex, but means they are not able to capture comparison and relation seeking tasks. In addition, through use of the quadrants, the framework in this thesis makes clear how the tasks can systematically be applied to different data items. The general mapping of Bach et al.’s tasks to the high-level categories of the classification developed in this thesis are shown in Table 25.

<table>
<tr><th rowspan="2"></th><th colspan="2">Lookup</th><th colspan="2">Comparison</th><th rowspan="2">Relation Seeking</th></tr>
<tr><th>Direct</th><th>Inverse</th><th>Direct</th><th>Inverse</th></tr>
<tr><td>Q1</td><td colspan="2">✓</td><td></td><td></td><td></td></tr>
<tr><td>Q2</td><td colspan="2">✓</td><td></td><td></td><td></td></tr>
<tr><td>Q3</td><td colspan="2">✓</td><td>✓</td><td></td><td></td></tr>
<tr><td>Q4i</td><td colspan="2">✓*</td><td>✓</td><td></td><td></td></tr>
<tr><td>Q4ii</td><td></td><td></td><td></td><td></td><td></td></tr>
</table>

Table 26 General mapping of Ahn et al.’s tasks to the high level categories of the classification developed in this thesis. Note that direct and inverse variations of lookup and comparison tasks are not explicitly considered. Comparison is considered at a high level, under compound tasks. Relation seeking tasks are not explicitly mentioned, nor are examples given. *Patterns discussed are more appropriate to Q3 tasks.

Table 26 shows the general mapping of Ahn et al.’s tasks to the classification developed in this thesis. The main limitation of Ahn’s work is the lack of a systematic explanation of the actual tasks involved. Their design space essentially specifies the data items which may be involved in tasks. As outlined in Section 2.4.3, they identify three aspects: entity (node/link, subgroup, network), property (structural or domain attributes) and temporal features (whether they occur at an individual time point or span multiple time periods (what they refer to as ‘aggregated events’). In the latter case, these can be described in terms of the shape of change (growth or contraction, convergence or divergence, stability, repetition, peak or valley) and rate of change (speed, acceleration or deceleration). These aspects can roughly be equated to the data model used in this thesis, with entities being the equivalent of references, properties being similar to characteristics, and temporal features spanning multiple time periods reminiscent of behaviours. One limitation of their framework (in data model terms) is that the shapes of change defined under temporal features (roughly equivalent to patterns) are too limited to adequately handle non-numeric attributes and are not appropriate to describe change in relational structures (see Section 5.2.1). Within the design space they map tasks which are described using terms such as ‘find’, ‘identify’, and ‘observe’, all of which can be considered to be look up or behaviour characterisation tasks (examples of both direct and inverse tasks are included, but these are not distinguished under their model).

During their evaluation of the taxonomy, Ahn et al. discovered they had omitted tasks involving comparison, correlation, and inference. While they seek to rectify this omission by adding a category of tasks called ‘compound tasks’, this category is limited to the occurrence of five types of tasks identified in the literature, and it is not entirely clear how these tasks combine with the dimensions identified in their design space. The five tasks they identify in this category can be mapped to either behaviour characterisation tasks or direct comparisons in quadrants 3 and 4i of the classification in this thesis. They do not consider comparison in quadrants 1 or 2.

	Lookup	Comparison	Relation Seeking
Q1	A, B, L	L	L
Q2	A, B, L	L	L
Q3	A, B	A	
Q4i	A, B	A	
Q4ii			
Connection Discovery			

Table 27 High level summary of task categories arising in extant works. A = Ahn et al. [41], B = Bach et al. [71], L = Lee et al. [40]

The above discussion has shown that with the exception of the two action tasks noted for Lee et al.’s classification, the tasks of the extant works can be captured by the task classification developed in this thesis. A very high level overview of the types of tasks mentioned in extant works mapped to the categories of the framework developed in this thesis is given in Table 27. We can see that relation seeking in quadrants 3 and 4 is not considered, nor are any tasks involving Q4ii (distribution of temporal trends over the graph). Additionally, while Ahn et al. did mention tasks involving inference and correlation amongst those they discovered to be missing, these do not appear in the list of compound tasks they include in their framework. None of the classifications therefore considered tasks associated with connection discovery. Note that the mapping offered here gives a very general overview and perhaps paints a rather generous picture of task coverage in extant classifications: it maps the case where an example of a task appears in a framework (as opposed to a category having been specified). It also does not highlight the lack of systematic coverage of task variations (such as inverse or direct variations of tasks) or variations in combinations of same/different time/graph objects in the other frameworks.

In terms of *comprehensiveness*, we can conclude that not only is the classification in this work able to capture all of the tasks of the extant frameworks, we can also see that none of the extant frameworks are able to capture all of the tasks of the other extant frameworks, and they also fall short in capturing the additional categories identified in this work.

In addition to demonstrating that the task classification is able to cover more task categories than any of the extant frameworks individually, it also gives us a little more information relating to the *real world nature of tasks*. Of the classifications reviewed, only Bach et al.'s took a formal approach to task generation. The others gathered tasks from the literature and extant systems: Lee et al. extracted their tasks from the literature (an extant taxonomy of tasks for tree visualisation and tasks used in user studies); Ahn et al. surveyed existing systems with regard to the tasks they support; Yi et al. appear to derive their tasks from their knowledge of social network analysis. Some evidence is therefore provided for the real world nature of the categories of the classification appearing in the extant classifications. Further investigation is needed to establish whether relation seeking in Q3 and Q4i, and tasks involving Q4ii and connection discovery, are indeed real world tasks or constructs of the formal process employed in the construction of the framework in this thesis.

6.2.2 Using the task classification in the design process

This section presents an empirical study which uses the developed classification as a generative method at the task understanding and abstraction stages of the design process. Further, the study demonstrates how the classification can be used to organise tasks and help establish which types of tasks are frequently occurring for a specific application. The use of task classifications in these scenarios was discussed in Section 2.3.4. As described earlier, this study is primarily designed to evaluate the *utility* of the classification in a usage scenario, but also offers further information relating to the *comprehensiveness of task coverage*, its *descriptive powers* and the *real-world nature of the tasks*.

6.2.2.1 Overview

The design of the study draws on the designs of the empirical evaluations carried out by Ahn et al. [41] and Schulz et al. [45]. This section offers a brief overview of the study design and how it seeks to evaluate the aspects outlined above. Further details of the design and implementation of the study follow in subsequent sections.

The chosen design scenario is that of developing a visualisation tool to help academics explore their department's co-authorship network in order to better understand collaborative working practices and publishing rates within their department. This scenario is in keeping with the examples used in Chapter 4 and Chapter 5, and the case study of Chapter 9.

The study was divided into two parts, both of which were conducted by email. In the first part, academics were presented with the analysis scenario and data set. They were asked to consider the data and note any questions which might be of interest to them. They were also asked to rate how interesting each of their questions were on a scale of 1-4 (where 1 was of least interest and 4 of most interest). The responses to these questions were then categorised using the high-level categories of the framework. Through this process, a number of task 'gaps' – task categories for which none of the participants had identified a task – were revealed.

For the second part of the study, a selection of the identified task gaps were presented to participants, and they were asked to rate how interesting they found them using the original scale, with the addition of 0 to indicate that a task was of no interest.

The study was specifically designed to evaluate the classification with respect to two usage scenarios:

[U1] Use of the classification as a generative method in the design process: as outlined in Section 2.3.4.1, one potential use of a classification is in overcoming the known problem of the difficulty of eliciting tasks by asking people to introspect on their task needs (a particular issue in Exploratory Data Analysis scenarios). Being able to find tasks of interest which participants had not previously considered by presenting them with a range of tasks generated using the classification evidences the utility of the classification in this usage scenario. In this case, the study is designed to answer the question: "*during requirements gathering, can the task classification be used to discover tasks of interest which have not previously been considered?"*

[U2] Use of the classification in task organisation: another important role that classifications can play during the design process is in characterising tasks in a consistent manner, and organising them to establish the most commonly occurring and important tasks. In this regard, the study seeks to answer the question: "*can the task classification act as a useful means of organising tasks?"*

In addition, the study evaluates the following properties of the classification:

[P1] Descriptive power and task coverage: part 1 of the study provides us with a set of real world tasks to be classified. The classification's ability to capture these tasks provides some evidence relating to its descriptive powers and task coverage. The study therefore helps us answer the question "*to what extent is the taxonomy able to capture real world tasks?".* Note that only a partial answer to this question is possible within the limits of the study, as the set of tasks is drawn from a single domain and analysis scenario; even if it is able to capture all of the tasks generated in the study, it is possible that tasks which cannot be captured may exist in other domains or analysis scenarios.

[P2] Real world nature of tasks: the set of tasks generated by participants in part 1, along with the set of tasks which participants find interesting in part 2, are examples of real world tasks. Categories of the classification to which these tasks can be mapped can therefore be said to be real-world in nature, rather than mere artefacts of the formal process followed in generating the classification. The study therefore goes some way to answering the question "*to what extent are the tasks of the taxonomy 'real world' (as opposed to artefacts of the formal process used in its development)?".* Again, the study may only partially answer this question due to the single domain and analysis scenario used. In the case that no tasks are mapped to a task category, we cannot conclude that this category is an artefact of the construction process, as examples may exist in other domains or analysis scenarios.

Finally, the study offers the opportunity to compare the developed framework with extant frameworks:

[CEx] Comparison with extant classifications: in the case that a task category which is not covered by extant classifications (see Section 6.2.1) and for which a real world task example is found during the study, the utility of using the classification developed in this work over extant frameworks (in terms of task coverage) can be demonstrated. In this respect, the study helps answer the question "*is there any advantage to using the developed classification over extant frameworks*?"

6.2.2.2 Study details

Pilot

The study was piloted with two subjects prior to running, and appropriate adjustments were made to the study design (see discussions relating to data, below, and identifying task gaps in Section 6.2.2.3). The results of the pilot study are excluded from the results reported here.

Participants

The participants - domain experts - were academics belonging to the Institute of Informatics and Digital Innovation (IIDI) at Edinburgh Napier University. 19 academics were invited to participate in the study, of which 12 accepted.

Data

Participants were provided with a data set consisting of publications data relating to approximately two-hundred authors and nearly two thousand publications, spanning a period of over thirty years. A description of the data, and an illustrative excerpt from the data, were included in the instructions to participants (Appendix C), along with web links to the full data set. The following data was made available:

Authors:

- Name
- Research centre affiliation (CAVES, CCER, CDCNS, CID, CSI)
- Joining and leaving dates

Publications:

- The list of authors
- The year in which it was published
- The type of publication (conference proceeding, journal article, book chapter, etc.)

To illustrate, an extract of the data is included in Table 28 and Table 29 below.

Table 28 Authors

Name	**Research Centre**	**Joined**	**Left**
Alan Cannon	CAVES	2003	-
Kevin Chalmers	CAVES	2005	-
Paul Craig	CAVES	2008	2012
Martin Graham	CAVES	1998	2015
Jessie Kennedy	CAVES	1991	-
Natalie Kerracher	CAVES	2010	-
Robert Kukla	CAVES	1996	-
Paul Shaw	CAVES	2008	-
Alistair Thomson	CAVES	2012	2013
…	…	…	…

Table 29 Publications

ID	Year	Authors	Type
1456	2015	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers	Journal Article
1455	2015	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers, Martin Graham	Conference Paper
1444	2014	Jessie Kennedy, *Externals*	Book Chapter
1401	2014	Martin Graham, Jessie Kennedy	Journal Article
1385	2014	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers	Conference Paper
1343	2014	Jessie Kennedy , *Externals*	Journal Article
1341	2014	Paul Shaw, Martin Graham, Jessie Kennedy, *External*	Journal Article
1248	2013	Paul Craig, Alan Cannon, Robert Kukla, Jessie Kennedy	Journal Article
1219	2013	Jessie Kennedy, Martin Graham, *Externals*	Conference Paper
1107	2013	Alistair Thomson, Martin Graham, Jessie Kennedy	Conference Paper
…	…	…	…

A decision was taken following initial piloting of the study to limit the data made available to participants to that relating to types and amounts of publications (as outlined above). In the initial pilot, the full details of each publication were given (from which research topic could potentially be extrapolated). However, this prompted a large number of questions which could not be answered directly based on this data (for example "*Are there any related spin off projects that appeared from a particular research over the years, or other extensions of the same work?*"). Whilst this would be an invaluable finding were the purpose of the study to actually design a visualisation system, it was not helpful when trying to evaluate the usefulness of the classification in task discovery (establishing whether the correct data is being used - whilst a key part of the domain problem characterisation stage of the design process - is not an intended use of the taxonomy). A decision was therefore taken to reduce the scope of the data in order to reduce the amount of responses which required data outside of that with which the participants were being presented. It appears that this had the unfortunate consequence of making the dataset less interesting to some of the participants (see *Participants' interest in the data* in Section *6.2.2.3*, below).

Instructions to participants

In the first part of the study, participants were presented with a real world design scenario. They were told that IIDI is developing a visualisation system to help people working within the Institute better understand its collaborative working practices and publishing rates. It was explained that as part of the design process, we want to find out what questions people using the visualisation system would like to be able to ask of the data that we have available. The data was presented, as outlined above, and it was explained how a co-authorship network could be constructed from this data.

Participants were first asked a multiple choice question relating to the capacity in which they might be interested in the data. Response options given were:

- In a management capacity
- Understanding my own data, e.g. looking at my own publishing track record, comparing myself with colleagues etc.
- Finding potential collaborators
- Understanding the data relating to my research group
- Other (please specify):

Participants were then asked to spend around 10-15 minutes considering the data, and note any questions which might be of interest. They were also asked to rate how interesting each of their questions were on a scale of 1-4, as follows:

1 = slightly interesting
2 = moderately interesting
3 = very interesting
4 = extremely interesting

In part 2, participants were presented with a list of tasks (described as "questions" in the information sheet) which were generated using the task framework and based on the task categories identified as gaps in part 1. They were asked to rate each task in terms of how interesting it was to them using the same scale as in part 1, with the addition of "0 = of no interest". If a participant did not understand a question, they were given the option to contact the evaluator for clarification, or note "DNU" (do not understand) in the relevant box. Illustrations were used in order to help participants understand the meaning of the tasks presented. The instructions made clear that these images were for illustrative purposes only, were constructed using synthetic data, and that there may be other, more appropriate ways to visualise the data to support a particular task.

The instruction sheets for parts 1 and 2 are included in Appendices C and D.

Task categorisation

For the purposes of the study, the participants' tasks returned in part 1 were categorised according to the following dimensions of the framework: task type (direct/inverse lookup, direct/inverse comparison, relation seeking), data quadrant (Q1, Q2, Q3, Q4i, Q4ii), and whether they involved attribute only, attribute and graph structure, or graph structure only. While further subcategorisation by variations within task types (such as additional constraints in inverse lookup tasks or whether comparisons involve the same or different times etc.) is possible and potentially useful at later stages in the design process, (for example, when selecting specific visual techniques), for the purposes of task discovery, a more general classification by data and task type was preferred when trying to establish the main aspects of the data and tasks in which people are interested.

One additional categorisation which was not considered when developing the task framework was made. This involved classifying tasks according to whether the data was aggregated on time and/or graph, and is discussed further in Section 6.2.2.4.

6.2.2.3 Results

Participants' interest in the data

In response to the question relating to participants' interest in the data, of the options offered:

- Half (6/12) of participants were interested in "*understanding my own data*"
- Around forty percent (5/12) were interested in "*understanding the data relating to my research group*"
- One third (4/12) were interested in "*finding potential collaborators*"
- One quarter (3/12) were interested in the data "*in a management capacity*"
- Three participants cited other reasons, including: *supporting researchers to find potential collaborators*, *understanding who the 'real experts' are*, and "*nosiness*".

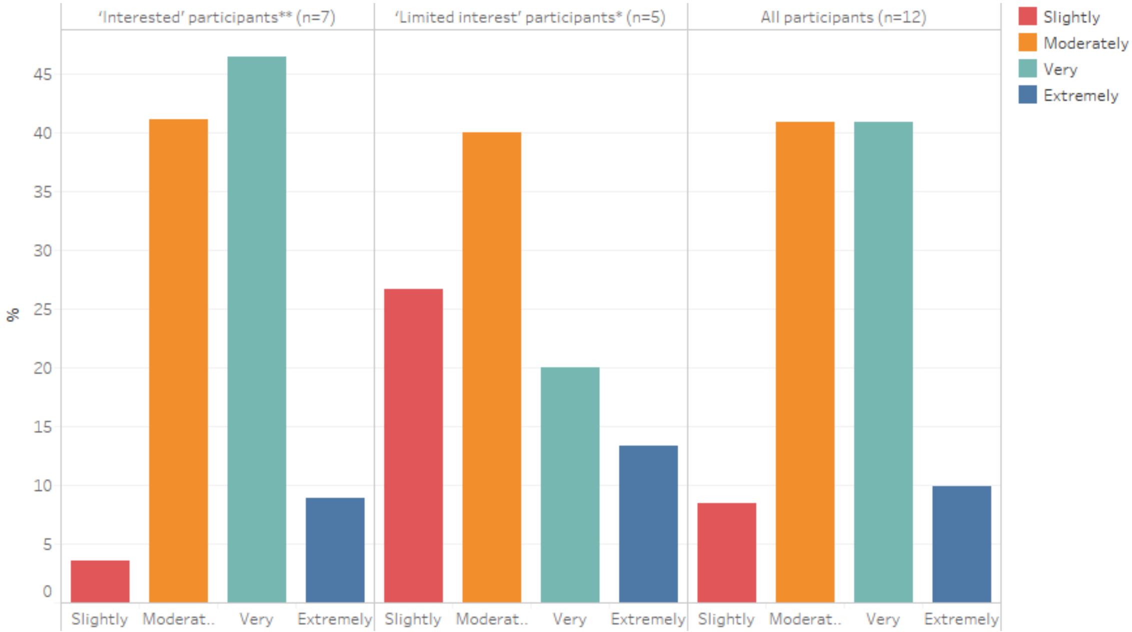
As noted in Section 6.2.2.2 (*Data*), above, following piloting, the dataset had been purposefully constrained in order to reduce the number of responses which required data outside of that with which the participants were presented. However, five out of the twelve participants explicitly commented that the data used in the study was of limited interest to them. The main reason given was that it lacked information on research topics and publication quality. One of these participants suggested that the data used would likely be of more interest to those working in a management capacity; interestingly, none of the five had selected this option. Of these five participants, two did not supply any questions of interest.

Tasks identified by participants

A total of 72 questions were returned by the 12 participants (mean = 6; max = 12; min = 0). Just over half of these questions (36/71[29]; 51%) were rated as very or extremely interesting. Note that the participants who explicitly stated their lack of

[29] One question was not rated.

interest in the data ("limited interest participants") returned fewer questions (mean of 3.2, compared to a mean of 6 per participant overall, and 8 per participant in the "interested participant" group). They also generally rated them to be of less interest, with only one third of questions (5/15[29]; 33%) rated as very or extremely interesting, compared to over 50% (31/56; 55.36%) in the interested participant group (Table 30).



	Total questions returned [*mean*]	Interest Rating (count)				
		slightly	**moderately**	**very**	**extremel y**	**not rated**
All participants (n=12)	72 [6]	6 (8.45%)	29 (40.85%)	29 (40.85%)	7 (9.86%)	1
'Limited interest' participants (n=5)*	*16 [3.2]*	*4 (26.67%)*	*6 (40%)*	*3 (20%)*	*2 (13.33%)*	*1*
*'Interested' participants** (n=7)*	*56 [8]*	*2 (3.57%)*	*23 (41.07%)*	*26 (46.43%)*	*5 (8.93%)*	*-*

**Participants who explicitly stated limited interest in the data*
***All other participants excluding not interested participants*

Table 30 Number of questions returned by participants and reported interest ratings. Limited interested participants are the five who explicitly indicated their lack of interest in the data. The remaining participants are considered to be interested participants.

Mapping participant tasks to classification

Appendix E includes the list of participant tasks, their categorisation under the task framework and, where necessary, explanations of how the categorisation was reached. Table 31 gives a summary of the numbers of participant tasks mapped to each of the categories of the classification.

		Direct Lookup/ Behaviour Characterisation	Inverse Lookup/ Pattern Search	Direct Comparison	Inverse Comparison	Relation Seeking
Q1 (node/edge at timepoint)	-	1	10 (all auxiliary tasks)			3
Q2 (graph at timepoint)	*Structure*					
	Attribute only					2
	Attribute + Structure					
Q3 (node/dyad over time)	*Structure*	2				
	Attribute	6	6			
Q4i (graph over time)	*Structure*	5	5			
	Attribute only					
	Attribute + Structure					
Q4ii (set of temporal trends)	*Structure*					
	Attribute		1			
Q4ii (distribution of temporal trends over the graph)	*Structure*					
	Attribute		1			

Tasks involving aggregated data		Direct Lookup/ Behaviour Characterisation	Inverse Lookup/ Pattern Search	Direct Comparison	Inverse Comparison	Relation Seeking
Q2 aggregated on graph	*Attribute*	3				
Q3 aggregated on time	*Attribute*	6	3	1		
Q4 aggregated on time	*Structure*	3	3	3		
	Attribute + structure	1	4			
	Attribute only	2	2			
Q4 aggregated on graph	*Attribute*	1	1			
Q4 aggregated on time and graph	*Attribute*	4		2		

	Structural comparison	Structural relation seeking
Q1	2	6 (1 auxiliary task)

Connection Discovery:	
Relationship between network structure and attributes	1
Relationship between structures	
Relationship between attributes	

Table 31 Number of participant tasks mapped to each category of the classification

Eight tasks were not mapped to the categories of the classification (see Table 32). The reasons for this were:

- Task did not make sense (2 tasks)
- Task involved an attribute not included in the data (4 tasks)
- Task was specified at a higher level of composition than that offered by the classification (2 tasks)

Task	Reason for exclusion
What is the ordering of people when the number of collaborators? (would be better if the external collaborators were known and so could be distinguished)	Doesn't make sense
How many times have 2 individuals published together for the first time?	Doesn't make sense
Do patterns of collaboration vary according to job status?	Job status does not appear in the data
What topic is X working on? (I didn't see it in the data, but presumably the publication reference must be available in the database, or at least the title? If it's not, feel free to discard this question)	Research topic does not appear in the data
What is the evolution of research topics for an individual/group over time?	Research topic does not appear in the data
Who else is publishing in journals that interest me	Journal details does not appear in the data
Is it possible to identify mentorship relationships in the data?	High level task
High level questions: • Who would I be able to help? • Who would be interested in me? • Who do I need to make friends with? ☺	High level task

Table 32 Tasks excluded from mapping

Identifying task gaps

As can be seen from Table 31, using the classification, it is possible to identify a number of categories of tasks which were not considered by participants. For each gap, an appropriate generic task description was constructed, along with two or three illustrative concrete examples. For example, for Q1 Direct Comparison, participants were asked: "Would it be interesting to compare attribute values between authors or between years? E.g. *compare Author A's publication count in 2015 and 2016; compare author A and author B's publication counts in 2015; compare author A's journal publication count in 2015 with their conference paper count."* Where appropriate, images were also used to help describe what was intended by the task description. For example, Q2 Behaviour Characterisation was illustrated with an image showing small multiples of a network changing over time, with a call out containing an enlarged image of the network in an individual year, to indicate that the task involved the network in a specific year of interest (Figure 55). Encodings were described where necessary. As noted previously, participants were instructed that these images were used to help illustrate the questions only, and had been constructed using synthetic data. They were also told that there may be other, more appropriate ways to visualise the data when answering a particular question.

Figure 55 Example of illustration used to assist with task understanding (Q2 Behaviour Characterisation)

Once task examples had been constructed for each of the identified gaps, the second part of the study was piloted. The instruction sheet and task examples used in the pilot are included in Appendix F. It became apparent during piloting that the amount of time required to consider every task gap in turn was far more than could reasonably be expected of our volunteer participants (estimated at over 4 hours; the pilot session was abandoned half way through when it became clear that it was taking an unreasonable length of time for the participant to complete.) Whilst such a process could justifiably be used in a real world requirements gathering process, for the purpose of evaluating the classification, it was decided to limit the number of tasks presented to participants.

The set of gaps involving behaviour characterisation and connection discovery were selected for use in Part 2 of the study. These are summarised in Table 33, and the participant instruction sheet containing task wording and examples is included in Appendix D. Not only did this provide a manageable number of tasks (16 tasks), this meant it was possible to explore whether all of the different aspects of the data were of interest. This was particularly important, as the one of the most fundamental differences in visual techniques (e.g. layout) required to support tasks comes from the different aspects of the data being explored (see Chapter 8). Further, it is likely that comparison or relation seeking in these quadrants is less likely to be of interest where behaviour characterisation is not of interest, therefore it makes sense to first ask about quadrants.

Task type (and attribute/pattern)		Question
Q2 (graph at timepoint) Behaviour Characterisation	structure	1.I
	structure & attribute (publication count)	1.II
	structure & attribute (research centre affiliation)	1.III
	attribute only (frequency distribution)	1.IV.a
	attribute only (ranking)	1.IV.b
Q4i (graph over time) Behaviour Characterisation	structure & attribute (publication count, research centre)	2.I
	attribute only (frequency distribution over time)	2.II.a
	attribute only (ranking over time)	2.II.b
Q4ii (set of temporal trends) Behaviour Characterisation	attribute (publication count)	3.I
	attribute (research centre affiliation)	3.II
	structure	3.III
Q4ii (time over graph) Behaviour Characterisation	attribute	4.I
	structure	4.II
Connection Discovery	between attributes (heterogeneous behaviours)	5.I
	between structure and attributes	5.II
	between structures	5.III

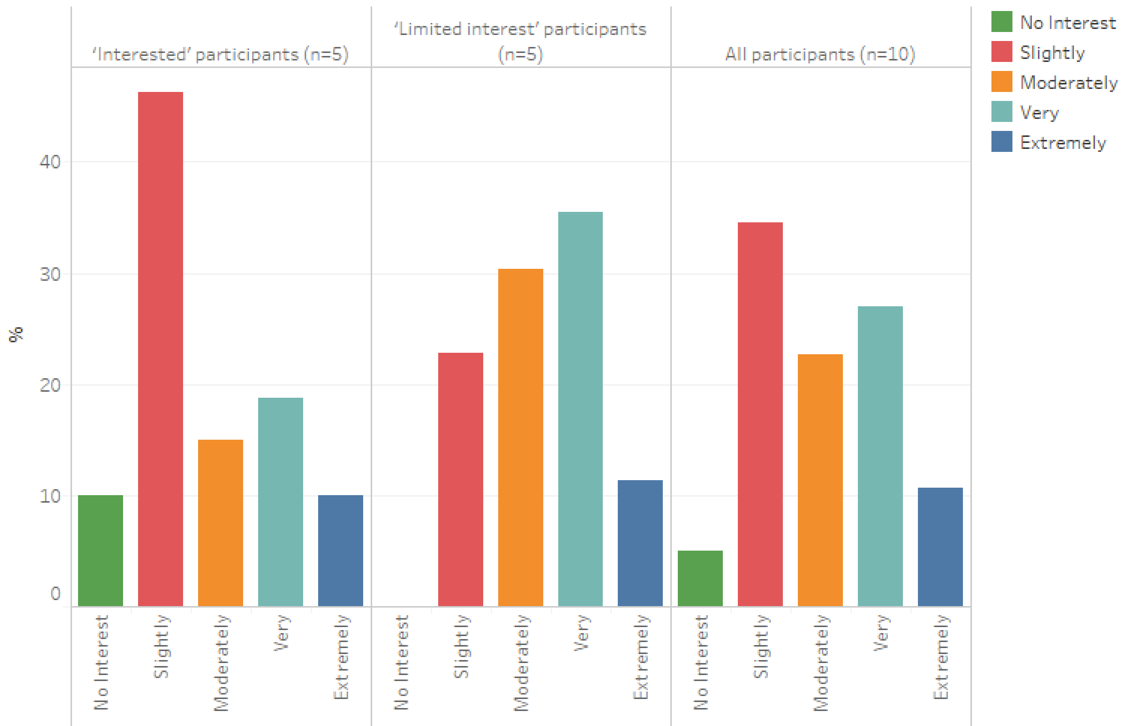
Table 33 The 16 task gaps investigated in Part 2. Examples of each type of task can be found in Appendix D. Tasks highlighted in blue are those which are included in this framework but are not found in the other extant task classifications (as discussed in Section 6.2.1)

Response to Part 2

Ten of the original twelve participants completed part 2 of the study, as two participants were not available (both of whom belonged to the 'interested' group of participants described in part 1).

Of the 16 tasks, all were found to be of some level of interest to the participants collectively. Overall, of the 159[30] ratings returned, over one third (38%) were ratings of very or extremely interesting. Only 8 ratings (5%) of no interest were returned. A difference in interest levels between the two groups of participants (interested and limited interest participants) distinguished in part 1 of the study can be seen in Figure 56. 47% of the limited interest group's ratings were very or extremely interesting, compared to 29% in the interested participants group. All "no interest" ratings were returned by the interested participant group. This finding was unexpected and is discussed further in Section 6.2.2.4.

[30] The 16 tasks were rated by each of the 10 participants; one participant omitted to rate one of the tasks.



	Total ratings	Interest rating none	slightly	moderately	very	extremely
All participants (n=10)	159*	8 (5.03%)	55 (34.59%)	36 (22.64%)	43 (27.04%)	17 (10.69%)
'Limited interest' participants (n=5)	*79**	*0*	*18 (22.78%)*	*24 (30.38%)*	*28 (35.44%)*	*9 (11.39%)*
'Interested' participants (n=5)	*80*	*8 (10%)*	*37 (46.25%)*	*12 (15%)*	*15 (18.75%)*	*8 (10%)*

*note that one participant omitted to rate one of the tasks

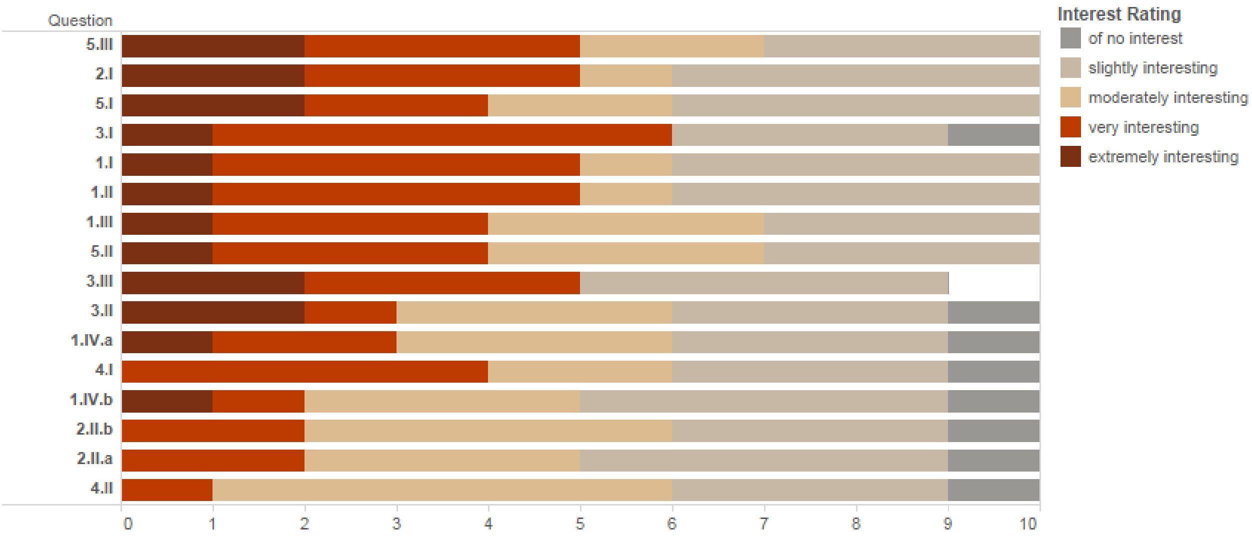
Figure 56 Overall ratings returned by 10 participants relating to how interesting they found 16 suggested tasks; also shown is the split by interest level in the data expressed by participants in part 1 of the study.

When considering task ratings at an individual task level, as would be expected, some tasks were found to be more interesting than others. Figure 57 shows the tasks ordered[31] by descending levels of interest. We can see that all tasks are thought to be of some level of interest. We can also see that the eight "no interest" ratings returned were spread over eight separate tasks (as opposed to being directed at a

[31] The tasks were ordered using weightings on the interest levels (no interest = 0, extremely interesting = 4) to calculate their position.

single task of very limited interest to participants). Over a third of the tasks were thought to be very or extremely interesting by half of the participants.

Of particular interest were connection discovery tasks between network structures (e.g. investigating the relationships between the structures of the co-authoring network at different time points, or whether changes in one part of the network affect other parts of the network), behaviour characterisation tasks involving changes in both structure and attribute values in the graph over time (e.g. how the network's structure and distribution of publication counts or research centre affiliations change over time), and connection discovery between attributes (for example correlation or influence between attributes, such as the relationship between research centre and publication counts). As can be seen in the table in Figure 57, it would not have been possible to find two of the tasks thought to be most interesting to participants. This is discussed further in Section 6.2.2.4, point [CEx].



Question	Task type (and attribute/pattern)	
5.III	Connection Discovery	between structures
2.I	Q4i (graph over time) Behaviour Characterisation	structure & attribute (publication count, research centre)
5.I	Connection Discovery	between attributes (heterogeneous behaviours)
3.I	Q4ii (set of temporal trends) Behaviour Characterisation	attribute (publication count)
1.I	Q2 (graph at timepoint) Behaviour Characterisation	structure
1.II	Q2 (graph at timepoint) Behaviour Characterisation	structure & attribute (publication count)
1.III	Q2 (graph at timepoint) Behaviour Characterisation	structure & attribute (research centre affiliation)
5.II	Connection Discovery	between structure and attributes
3.III	Q4ii (set of temporal trends) Behaviour Characterisation	structure
3.II	Q4ii (set of temporal trends) Behaviour Characterisation	attribute (research centre affiliation)
1.IV.a	Q2 (graph at timepoint) Behaviour Characterisation	attribute only (frequency distribution)
4.I	Q4ii (time over graph) Behaviour Characterisation	attribute
1.IV.b	Q2 (graph at timepoint) Behaviour Characterisation	attribute only (ranking)
2.II.b	Q4i (graph over time) Behaviour Characterisation	attribute only (ranking over time)
2.II.a	Q4i (graph over time) Behaviour Characterisation	attribute only (frequency distribution over time)
4.II	Q4ii (time over graph) Behaviour Characterisation	structure

Figure 57 Part 2 tasks: stacked bars show count of interest ratings for each task, in descending order by interest level. Concrete examples of each type of task can be found in Appendix D. Tasks highlighted in blue are those which are included in this framework but are not found in the other extant task classifications (as discussed in Section 6.2.1)

6.2.2.4 Discussion

Let us consider the results of the study in relation to the questions outlined in Section 6.2.2.1.

[P1] *Descriptive power and task coverage: To what extent is the taxonomy able to capture real world tasks?*

The set of 72 tasks returned by participants in part 1 of the study were classified according to the categories of the framework. All tasks - with the exception of the eight discussed in Section 6.2.2.3 – were successfully classified. The inability to classify the eight tasks did not suggest a need to extend the taxonomy with additional categories; the reasons for the difficulty in classifying stemmed largely from issues with the tasks themselves (either tasks which did not make sense or involved attributes that did not exist in the data). Only two tasks could not be classified as they were specified at a higher level of composition than that offered by the classification. These tasks could have been categorised following further specification by the participants (the task taxonomy could potentially be useful in this regard in exploring decomposition with the participant), but for the purposes of the study they were omitted. Based on this, it can be concluded that the taxonomy was successfully able to capture this specific set of real world tasks, and – within the limits outlined in Section 6.2.2.1 – this provides evidence in favour of the taxonomy's descriptive abilities.

The process of abstracting tasks was non-trivial and required an iterative process to ensure that tasks were consistently categorised. A number of choices needed to be made during the classification process, which are discussed further in this section. This was in a large part due to the difficulties associated with translating vague natural language into precise formal definitions. For example, task 35, "*who is still currently in the School?"* could potentially be translated in a general way as a question asking about which members of staff are currently in the School (direct lookup), or it could be translated more precisely as a task which asks about members of staff in the current year who were also present in the previous year (relation seeking). Where a more precise translation was available, this was selected.

As noted by Andrienko, it is often possible to describe tasks either as a sequence of elementary tasks or as a single synoptic task. For the purposes of this study, preference was given to synoptic description. This is not only because synoptic tasks are given primacy in the framework, but because the different quadrants - which reflect the different possible data items on which synoptic tasks operate - require markedly different visual techniques.

Many tasks made no reference to a specific year or a particular period of time. For example, task 64[32], "*Who's working with whom*?", or task 21, "*Who is collaborating without external partners?*". These questions could potentially be asked of a specific year, or as an aggregation of the whole time period (or a subset of the time period). Similarly, some tasks considered the set of authors together as a whole, for example, task 49, "*How many papers of a particular type were published in year X?*" – such a question could be asked of an individual author or the set (or subset) of authors. Finally, some tasks displayed both of these features – making no mention of time and considering the whole set of authors together - for example task 51a. "*What's the average publication rate?".* All of these questions involve some sort of summary statistic, such as an average value or an aggregated total (over time and/or for the set of authors in the graph). Under the Andrienko framework, these are simply treated as a pattern at a very high level of granularity (e.g. the pattern of an attribute over time could be described in detail including all changes in the trend; in less detail as e.g. 'an increasing trend'; or using some summary statistic such as aggregate, mean, median, mode, highest/lowest values, most frequent value (for categorical values), etc.). Patterns at these different levels of granularity may involve very different techniques. As the Andrienko framework does not attempt to relate

[32] see Appendix E for categorisation of participants' tasks

techniques directly to task categories, they do not discuss this further. However, as one of the intended purpose of developing the taxonomy and visual technique mapping in this thesis is to help in the appropriate selection of suitable techniques, these cases were given further consideration. It became clear from reviewing these tasks that we can aggregate (summarise) in three ways: on time, on graph, or on both time and graph. Interestingly, we do not need a new set of techniques to cover these cases; aggregating on the temporal and/or graph dimension simply reduces these tasks to those of the quadrants which deal with single time points or individual nodes, as follows:

- Q2 aggregated on graph = elementary task (a single value is used to represent an attribute associated with the set (or a subset) of nodes at a single time) = graph or subgraph treated as a single reference (node) + time point
- Q3 aggregated on time = elementary task (a single value represents an attribute's value over a time period for an individual author) = time period treated as a single reference (time point) + individual node
- Q4 aggregated on time = Q2 task (the graph is flattened into a single graph and individual values represent each node's/edge's attribute values over a time period) = time period treated as a single reference (time point) + graph
- Q4 aggregated on graph = Q3 task (a single value is used to represent an attribute associated with the set (or a subset) of nodes at each time point) = graph or subgraph treated as a single reference (node) + time period
- Q4 aggregated on time and graph = elementary task (a single value represents the attribute values associated with all nodes (or a subset) and all time points (or a time period)) = graph or subgraph treated as a single reference (node) + time period treated as a single reference (time point)

Note that while for the purpose of *selecting techniques* these tasks would best be grouped together with their equivalent (reduced) quadrant, for the purposes of *identifying gaps* in the tasks identified by participants, these were grouped together with the original (unreduced) quadrant, in order to better reflect participant intention (i.e. where their question related to the whole graph at an individual time, not to an individual element, these would be coded as Q2, even though a technique applicable to elementary tasks would be an appropriate choice to support the task).

One final interesting point which arose during task translation was the case of a hybrid direct lookup/behaviour characterisation and inverse lookup/pattern search task. One scenario in which this type of task arises is somewhat characteristic of exploratory data analysis: the case where 'I don't know what I'm looking for until I see it'. In this case, we first need to look at the data and characterise its behaviour (behaviour characterisation). In doing this, we might notice some (sub) patterns of interest. We would then investigate these further, finding out who and what times they are associated with, and perhaps looking for this noticed pattern in other parts of the network. However, this second part is not strictly pattern search as we did not start out with a pattern in mind that we were searching for. The second scenario that this hybrid arises in is questions such as "*When was the first paper published by X?*". While we might treat this as behaviour characterisation in Q3, we need to report the start time (the referential component) as a feature of this pattern, which is indicative of a pattern search task.

In translating the participant tasks, a decision was taken to translate these cases as behaviour characterisation in its most general sense, allowing referential components to feature as part of the characterisation of the pattern. However, these would perhaps be better characterised as 'pattern browsing' rather than simply characterising attributes associated with a specific referential component or explicitly searching for some pattern known a priori.

In summary, while task translation was not trivial, and a number of important considerations were noted during this process, all of the tasks (with the exception of those previously mentioned) could be captured by the high level categories of the framework, providing evidence in support of its descriptive powers and comprehensiveness.

[U2] *Use of the classification in task organisation: Can the task classification act as a useful means of organising tasks?*

To demonstrate the usefulness of the classification in task organisation, Table 31 - which shows participants tasks mapped to each category of the classification - has been reorganised in Table 34 to show quadrant plus task type for use in tool selection (as per the discussion relating to descriptive powers, above). Note that the found tasks from part 2 of the study have not been included here (recall that these were rated by each participant, therefore it would skew the tasks heavily in their favour if they were included). Also note that for simplicity, only frequency of occurrence is included in this discussion, although importance of tasks could potentially be factored in by weighting tasks according to their importance rating. The point of this discussion is simply to demonstrate the usefulness of the task classification in organising tasks.

	Direct Lookup/ Behaviour Characterisation	Inverse Lookup/ Pattern Search	Direct Comparison	Inverse Comparison	Relation Seeking	*Total*
Q1 (node/edge at timepoint) **+** **Q2 aggregated on graph** **+** **Q3 aggregated on time** **+** **Q4 aggregated on time and graph**	14	13	3		3	*33*
Q2 (graph at timepoint) **+** **Q4 aggregated on time**	6	9	3		2	*20*
Q3 (node/dyad over time) **+** **Q4 aggregated on graph = Q3**	9	7				*16*
Q4i (graph over time)	5	5				*10*
Q4ii (set of temporal trends)		1				*1*
Q4ii (distribution of temporal trends over the graph)		1				*1*
Total	*36*	*36*	*6*		*5*	*81*

	Structural comparison	Structural relation seeking
Q1	2	6 (1 auxiliary task)

Connection Discovery:	
Relationship between network structure and attributes	1
Relationship between structures	
Relationship between attributes	

Table 34 Organisation of participant generated tasks according to quadrant and task type. Light-dark green shading emphasises the number of tasks of each type (low-high).

From Table 34 we can see that the most common data items of interest are individual nodes/edges (not surprising, as tasks involving elements are often included as part of larger tasks), followed by tasks involving a single graph structure (either a graph at a particular point in time or the whole graph aggregated on time), then tasks involving time series (either for individual authors or some metric representing the whole graph). Of slightly less interest was the evolving graph over time, and much less interesting still is the set of temporal trends or distribution of temporal trends over the graph (note, however, that we found in part 2 of the study that these behaviours were of interest).

In terms of task type, lookup tasks are by far the most common, with some interest in comparison and relation seeking involving elements and individual graphs.

Structural relation seeking (finding nodes connected in a particular manner) is also of interest.

What this would suggest when selecting tools for an interface in this case (without further prompting) participant's questions more frequently relate to understanding and looking for patterns in the graph at individual timepoints or in the aggregated graph, and the temporal data associated with individuals/pairs of individuals, than they are in understanding temporal changes in the graph over time (or the set of temporal trends). This is particularly interesting as a typical temporal graph visual solution (such as a sequential or small multiple views of the graph evolving over time) may not be the best option for supporting tasks involving individual graph structures and individual temporal trends (this point is discussed further in Chapter 8). Considering tasks at a lower level of detail e.g. whether structure/attributes are of interest, could also provide us with further information upon which to make design decisions, and is explored further in the case study of Chapter 9.

[U1] *Use of the classification as a generative method in the design process: During requirements gathering, can the task classification be used to discover tasks which have not previously been considered?*

All of the task gaps identified using the classification were found to be of some level of interest to participants. At the most basic level, it can therefore be affirmed that the classification can be used to discover tasks that had not been previously considered. Moreover, at least one third of these tasks were rated as very or extremely interesting by at least half of the participants; this indicates that using the classification in this way can find not only tasks of passing interest to participants, but also those which could potentially be important to people carrying out an analysis. While the data collected in part 1 (individual interest ratings relating to participants' own questions) is not directly comparable to that of part 2 (all participants' interest ratings for the suggested questions), this figure looks respectable given that participants rated their own questions to be very or extremely interesting only in around half of cases. It should also be noted that participants were not asked how interesting they found *each other's tasks* in part 1, which could potentially have provided background information relating to agreement in levels of interest on individual tasks across the group as a whole.

One unexpected observation relating to the interest levels in the suggested tasks of part 2 is that those participants who expressed only a limited level of interest in the data in part 1 rated the suggested questions more highly in terms of interest than participants in the interested group (47% of ratings returned by the limited interest group were interesting or very interesting, vs 19% in the interested group). Further, only those in the interested group returned ratings of “no interest”. While further investigation is needed to explain this difference, one possibility is that the interested participants had a clearer idea of possible tasks at outset than those in the limited interest group, therefore the suggested tasks were rated less interesting as they had already articulated the tasks that were of most interest to them. Another is that those who were less interested in the data had not been able to anticipate the range of possible questions it might help them answer, reminiscent of the known difficulties in asking people to introspect and pre-empt their task needs in an Exploratory Data Analysis scenario.

Given that by using the classification it was possible to discover tasks of significant interest to participants which they had not previously considered, it can be concluded that the study has found evidence in favour of the usefulness of the classification as a generative method during requirements gathering. However, this should be qualified by the following points:

Usability: Not considered in this study was the ease of using the classification by visualisation researchers (other than the author of the classification). Firstly, using the classification requires familiarity with the classification and its terminology, particularly when translating tasks. Secondly, the study was specifically designed to demonstrate that the classification can be used to generate and discover tasks of interest, however, the approach adopted in the study would not necessarily be the most appropriate method in a real-world requirements analysis scenario. As found in part 2 of the study, a great many tasks can be generated using the classification, which potentially could result in a very large set of tasks on which to gain feedback, requiring much time and input from domain experts. Using the classification for task generation in a more flexible manner when requirements gathering in the real world would likely be a more appropriate approach. For example, considering the parameters in turn during discussions with experts would allow the researcher to narrow down those tasks of most interest e.g. they might initially establish which data items are of most interest before considering the variations of tasks in which these data items might be involved, in progressively greater levels of detail (thus avoiding the need for experts to individually consider and discount potentially hundreds of irrelevant tasks).

Comparison with other methods: The use of the classification was not compared to other generative methods (such as brainstorming or focus groups). Further work is needed to establish how using the classification to generate and discover tasks compares to other methods in terms of, for example, task coverage, time and effort involved, and user experience of the process.

[P*2]* *Real world nature of tasks: To what extent are the tasks of the taxonomy 'real world' (as opposed to artefacts of the formal process used in its development)?*

Participants' tasks which were returned in part 1 of the study were classified into a number of different categories, as presented in Table 31. A number of task gaps – where no participant task was mapped to the framework – were identified. These gaps consisted largely of comparison and relation seeking tasks. As discussed in Section 6.2.2.3, it was not possible to investigate all of these gaps within the constraints of the study, so only gaps in behaviour characterisation and connection discovery tasks were investigated. Examples of each of these types of tasks were either provided by participants themselves in part 1 of the study, or thought to be of interest to participants in part 2; this provides important evidence (within the limits outlined in Section 6.2.2.1) that all of the behaviour characterisation and connection discovery tasks are real world in nature.

Let us combine this finding with that of the discussion in Section 6.2.1 to gain a better idea of what can be said overall of the real world nature of tasks in the framework. In Section 6.2.1, as three of the extant task classifications used empirical methods to derive their tasks, evidence was provided for the real world nature of the categories of the classification which overlap with those of the extant classifications (see Table 27). The categories which were *not* covered by extant classifications are:

- Lookup, comparison, and relation seeking in Q4ii
- Relation seeking in Q3 and Q4i
- Connection Discovery

	Lookup	**Comparison**	**Relation Seeking**
Q1	A, L, ESp1	L, ESp1	L, ESp1
Q2	A, L, ESp1, ESp2	L, ESp1	L, ESp1
Q3	A, ESp1	A , ESp1	?
Q4i	A, ESp1, ESp2	A, ESp1	?
Q4ii	ESp1, ESp2	?	?
Connection Discovery	ESp1, ESp2		

Table 35 High level task categories which are represented in extant frameworks or were identified in the empirical study, and those requiring further investigation as to their real-world nature (highlighted in red). Key: A = appears in Ahn et al.'s framework; L = appears in Lee et al.'s framework; ESP1 = reported to be of interest in part 1 of the empirical study; ESP2 = found to be of interest in part 2 of the empirical study (note that Yi et al.'s framework is not shown as it could not be directly mapped to the categories of the framework; Bach et al.'s framework is not included as it is constructed using a formal process, therefore does not provide evidence in support of the real world nature of tasks).

Table 35 combines the tasks found to be real world via the extant frameworks along with those found to be real world via the study. Shaded in blue are tasks which were found to be real world in the study, but were not represented in extant works. Shaded in red are the tasks which were not covered by extant frameworks or reported in the study. This latter set of tasks (comparison in Q4ii and relation seeking in Q3, Q4i, and Q4ii) therefore require further consideration as to their real world nature.

As outlined in Section 6.1.3.3, evaluation of the real world nature of tasks can be tricky, and is often dealt with in the literature by providing illustrative concrete examples for each possible category of abstract task. Before moving on, let us here consider some examples of real world tasks drawn from the literature for those categories for which a question mark remains.

- **Comparison in Q4ii**: one example of making comparisons (and also finding relations) between the distributions of temporal trends in groups of nodes can be seen in the application of Burch and Weiskopf's [112] TimeEdgeTrees. Their application example discusses using their technique to inspect the water levels of 450 measurement stations of rivers in Germany, which form a natural hierarchy. They are interested in the water level movements and the water level minima and maxima over time, in particular if the water levels of river subsystems influence the water levels of the larger rivers. Part of their analysis requires them to compare the patterns of subsystems, i.e. the sets of temporal patterns of groups of rivers in the graph structure.

  Another example of comparison in Q4ii is that of Henry Riche et al.'s LinkWave [113]. They demonstrate application of their system in a neuroscience context. One task which their system is designed to support is comparison of the temporal trends between different connected groups of neurons in an individual's brain. Another is comparison of the set of temporal trends in neural connectivity associated with a healthy brain and that of a diseased brain.

- **Relation seeking in Q3**: Hocheiser and Schneiderman's [114] design studies demonstrate the use of their TimeSearcher tool in a biological context. One of the features of TimeSearcher (discussed further in Section 8.4) is its ability to allow searching for similar or opposite temporal trends to that of a selected trend. In their design study, they note the tool's use in searching for temporal trends in a microarray data set to find expression profiles similar to that of a gene which is known to be involved in cell death.

- **Relation seeking in Q4i**: Gloor and Zhao [75] use their iQuest system to investigate social communication networks in organisations as they change over time. Of interest to them are questions relating to the similarities and differences between the uses of different communication technologies in temporal networks, for example "*does the same group of people exhibit different network attributes when interacting via telephone, email, face-to-face or other".* Such a question can be considered to involve both comparison and relation seeking tasks in Q4i.

- **Relation seeking in Q4ii**: Saraiya et al. [115] abstract a number of general graph tasks from common needs in bioinformatics pathway analysis in order to evaluate their temporal graph visualisation system. One of their tasks, "*find a group of nodes that display most different behaviour than the rest of the graph over all the time points*", is a good example of a relation seeking task in Q4ii.

These illustrations of tasks from the literature provide some further evidence as to the real world nature of the tasks in the framework. Note, however, that this discussion has summarised the evidence at a high level, and further work at a finer level of granularity, to cover inverse and direct variations of tasks, and structure vs attribute, along with the further dimensions which were not explored in the evaluations (e.g. the extent to which data items are specified in tasks, or same/different time/graph components are involved in comparison tasks etc.) is also needed in order to confirm the real world nature of all the variations of tasks in the framework.

[CEx] *Comparison with extant classifications: Is there any advantage to using the developed classification over extant frameworks?*

Reflecting on the discussion in Section 6.2.1, the extant classifications do not cover relation seeking in Q3 or Q4i, any tasks involving Q4ii (distribution of temporal trends over the graph; set of temporal trends considered together), or connection discovery tasks. One advantage of using the framework in this thesis therefore is its ability to describe more tasks than those of extant works. This can be demonstrated firstly by considering the classification of the set of tasks identified by participants during part 1 of the study: two pattern search tasks involving quadrant 4ii behaviours were identified (task 43. *Years with the highest number of publications for each author, relative to joining the department. (Which career phase is most productive?),* and task 40. *Who are the most experienced researchers 'near' me in the network? (ie who could I go to for advice)*), along with one connection discovery task (task 1. *Whose publication rates have been affected by someone else arriving or leaving?*). The extant frameworks would have difficulty capturing these tasks within their categories.

Secondly, in part 2 of the study, the set of task gaps which were investigated included Q4ii behaviours and connection discovery, tasks which are not found in the extant task classifications. Not only were all of the suggested tasks associated with these aspects of the data thought to be of some level of interest by participants, connection discovery tasks between network structures and connection discovery between attributes were two of the most highly rated tasks in terms of participants' interest. It would not have been possible to "discover" these tasks using the extant frameworks.

If we consider the extant frameworks individually, they are likely to have performed significantly worse in terms of classifying and discovering tasks. Table 36,Table 37, and Table 38 show the tasks identified by participants in part 1 and tasks discovered in part 2 mapped to the high level categories of the framework. The blue shaded area shows the task categories *not* covered by the individual frameworks (as per discussion in Section 6.2.1.). We can see that individually, many types of tasks would be difficult to classify and/or discover using the frameworks individually.

	Lookup	Comparison	Relation Seeking
Q1	P	P	P
Q2	P, D	P	P
Q3	P		
Q4i	P, D		
Q4ii	D		
Connection Discovery	D		

Table 36 Lee et al.: tasks identified by participants in part 1 (P) and tasks discovered in part 2 (D) mapped to the high level categories of the framework. Blue shaded area indicates task categories *not* covered by Lee et al.'s framework.

	Lookup	Comparison	Relation Seeking
Q1	P	P	P
Q2	P, D	P	P
Q3	P		
Q4i	P, D		
Q4ii	D		
Connection Discovery	D		

Table 37 Ahn et al.: tasks identified by participants in part 1 (P) and tasks discovered in part 2 (D) mapped to the high level categories of the framework. Blue shaded area indicates task categories not covered by Ahn et al.'s framework.

	Lookup	Comparison	Relation Seeking
Q1	P	P	P
Q2	P, D	P	P
Q3	P		
Q4i	P, D		
Q4ii	D		
Connection Discovery	D		

Table 38 Bach et al.: tasks identified by participants in part 1 (P) and tasks discovered in part 2 (D) mapped to the high level categories of the framework. Blue shaded area indicates task categories not covered by Bach et al.'s framework.

However, it is interesting to note that if we take this high level view of task coverage and combine the three extant frameworks together, we see that the majority of tasks identified or discovered during the study correspond to those categories covered by the extant frameworks (Table 42). As noted in Section 6.2.2.3, it was not possible to investigate all of the task gaps (including comparison and relation seeking tasks) in part 2 of the study. While evidence from the literature for the real world nature of these tasks was considered in Section 6.2.1, in light of the strong correspondence between the task types identified by extant frameworks and those returned by the study (at least when viewed at this very high level), it would be very interesting to investigate these tasks further in terms of both their frequency of occurrence and real world nature.

	Lookup	Comparison	Relation Seeking
Q1	P	P	P
Q2	P, D	P	P
Q3	P		
Q4i	P, D		
Q4ii	D		
Connection Discovery	D		

Table 39 Extant frameworks combined: tasks identified by participants in part 1 (P) and tasks discovered in part 2 (D) mapped to the high level categories of the framework. Blue shaded area indicates task categories not covered by the three extant frameworks.

Of course, we should also bear in mind that this discussion has taken a very (perhaps over-) simplified view of the task categories, ignoring many of the important distinctions made in the framework which are not captured by the extant works (for example, further distinctions in task type and whether the data items participating in these tasks involve structure only, attribute only or attribute in a structural context). As we will see in Chapter 8, these distinctions are important when considering visual techniques for their support. This highlights one further – and perhaps the most important advantage – of using the classification proposed in this thesis: that a direct mapping between the tasks and the visual techniques which are able to support them is offered, allowing it to be used when selecting techniques during the design process. While both Lee et al. and Ahn et al. describe existing systems in terms of the tasks they support, they do not offer such an overview of visual techniques organised by the tasks which they are able to support, for use in the design process. Bach et al. do not consider visual techniques in relation to their tasks.

Finally, as discussed earlier, one aspect of the task classification which has not been evaluated is its usability. Ahn et al. is the only one of the three extant classifications to have been evaluated with respect to its usability. In their interviews with domain experts they found that they were "*neutral on ease of use*". While the classification in this work is intended to be used by visualisation researchers (as opposed to domain experts) and has shown to be more comprehensive than the extant works, it is arguably more complex. It would therefore be useful to evaluate it in terms of its usability in comparison to the extant works.

6.2.2.5 Conclusion

This study has demonstrated the utility of the classification in task generation, discovery, and organisation, during the visualisation design process. The study has also provided evidence in support of the descriptive powers of the classification. While the study suggests that the behaviour characterisation and connection discovery tasks are indeed real world in nature, further work is required to establish the real-world nature of some of the tasks in the classification. While the classification developed in this thesis can justifiably claim that it is more comprehensive than extant task frameworks, additional work is required to fully evaluate the usefulness of the classification in relation to extant classifications, particularly in terms of its usability during the design process.

6.3 **Summary**

This chapter has reflected on existing evaluation practices appropriate when evaluating a task classification. Four main aspects which can be evaluated were identified (evaluation of construction method; evaluation of a classification's properties; evaluation of usage; and evaluation with respect to adoption), and methods appropriate to evaluating each aspect were discussed. Based on this research, the task classification in this work was evaluated firstly in relation to extant temporal graph task classifications with respect to the properties of comprehensiveness and descriptive powers, and secondly in an empirical study primarily designed to assess its utility in the design process.

While further work remains to determine the extent to which the tasks of the framework are real-world in nature and its usability by visualisation researchers, clear evidence in favour of its comprehensiveness and descriptive abilities were shown both in comparison to extant frameworks and in the empirical study. The empirical study demonstrated the usefulness of the classification in both task discovery and organisation. The use of the framework in the evaluation process is explored further in the case study of Chapter 9.

Chapter 7 Visual techniques for temporal graph data: a design space

This chapter considers the visual techniques for representing temporal graph data. It discusses the development of a design space of visualisation techniques for temporal graph data, which brings order to the existing work in the area, and is used to identify possibilities for new techniques. Specifically, this chapter:

- Reviews existing work relating to visual techniques for temporal graph visualisation, and classifications of these techniques.
- Identifies two dimensions upon which the visual techniques can be classified and combines these dimensions to produce a design space.
- Maps existing techniques to this design space
- Identifies gaps in this design space, which may prove interesting opportunities for the development of novel techniques.

The chapter also includes a discussion of the relative strengths and weaknesses of the different possible encodings identified when constructing the design space.

The chapter is organised as follows: Section 7.1 discusses the methodology adopted. The literature that was reviewed to extract the dimensions of the design space and the development of the categories within each of these dimensions is also discussed. Section 7.2 presents the structure of the design space. The mapping of existing techniques to the design space and related findings are presented in Section 7.3. Finally, Section 7.4 discusses the strengths and weaknesses of the encodings identified when constructing the design space.

7.1 Developing the design space

As discussed in Section 2.4, several works in the Information Visualisation literature have focussed their attention on categorising existing visualisation techniques. While some work specific to classifying temporal graph visualisation approaches has already been carried out, the ‘space of the possible’ has not yet been explored. In order to address this, a design space of temporal graph visualisation techniques was constructed.

A design space can be constructed by combining the independent dimensions of a taxonomy to produce all possible variants. The first step in constructing the design space, therefore, was to identify the independent dimensions. In order to identify these dimensions, existing classifications of temporal graph techniques were reviewed, from which two distinct dimensions emerged: temporal and graph structural encoding.

The next step was to identify the distinct categories within each dimension. In addition to the categories identified in the existing classifications, the systems and techniques literature was surveyed to identify any further categories. Once the dimensions and categories within each dimension were established, a matrix was constructed, into which the existing techniques were organised.

7.1.1 Background: Existing classifications of visual techniques

A number of surveys and classifications of visual techniques have been reviewed in order to extract the dimensions upon which temporal graph visualisation techniques can be categorised. Included were surveys and categorisations of graph visualisation techniques e.g. [79], [80], [116], [117], a number of which focus on hierarchical structures e.g. [38], [81], [118]. Also considered were the papers relating to the categorisation of visualisations of temporal data, such as [3], [98], [119]–[121]. Of particular interest from outside of the graph and temporal graph visualisation domain were Javed and Elmqvist's [39] design space of composite visualisations and Gleicher et al.'s [102] taxonomy of techniques for visual comparison.

As briefly outlined in Chapter 2, some discussion exists in the literature with specific regard to classifying visual approaches for temporal graph data. For example, Hadlak et al. [82] categorise visual approaches for large dynamic graphs based on the reduction techniques used: whether the temporal or structural element of the graph is reduced, and whether the reduction is via abstraction or selection, or is unreduced. Combining these dimensions results in a 3x3 matrix (structure v time; unreduced;

reduced by abstraction; reduced by selection). Federico et al. [28] divide their discussion of possible representations with respect to the mapping of the temporal dimension (mapping to time = animation; to space = juxtaposition; to a visual variable = superimposition; to an additional spatial dimension = 2.5D). Rufiange and McGuffin's [83] taxonomy is also based on the temporal dimension, dividing the techniques into small multiples, animation, embedded glyphs, linearised graph plus time axis, and 3D. von Landesberger et al. [79] classify graphs according to their time dependence (static vs time-dependent; with further subdivision of time-dependent graphs based on whether they involve attribute change, structural change, or both) and graph structure (trees, generic graphs, and compound graphs). Recently, Beck et al. [84] surveyed the existing approaches for temporal graph visualisation and produced a hierarchical taxonomy of dynamic graph visualisation techniques. At the top level, they distinguish animated and timeline approaches for temporal encodings. Animated approaches are further subdivided by the layout algorithm used, while further sub-categorisations of the timeline category are made according to temporal and graph structural encodings used.

In all of these discussions, a key distinction between the temporal and graph structural dimensions is apparent. This is therefore used as the fundamental division to construct the design space, which shows the possible combinations of graph structural and temporal encodings.

7.1.2 Dimensions of the Design Space

Two independent dimensions upon which visual techniques for temporal graph data can be classified were identified in the literature: graph encoding and temporal encoding. The possible categories along each of these dimensions are now considered.

7.1.2.1 Graph dimension

There is a huge amount of literature relating to static graph visualisation [69]. The key challenge focusses on laying out the graph to represent relations between elements in a readable manner - affording the viewer an accurate, usable, and readily understandable, representation of the graph's structure - while being computable in an acceptable timeframe. As more than one layout can correspond to the same graph structure, a set of aesthetic criteria [14], [122] along with numerous layout algorithms have been developed. The difficulties for graph layout are compounded at scale, and recent work has focussed on the problem of visualising large graphs e.g. [79], [82]. An additional challenge is that of multivariate graphs. While much of the focus for graph drawing has been on representing the graph's topological structure, an additional problem is finding suitable ways to represent multiple node and edge attributes. Having used up the spatial dimensions for graph layout, possibilities for attribute representation are restricted. Moreover, we often wish to represent attribute values in the graph context, thus the tiny amount of space available to represent each node and edge's attribute values is a major issue.

The underlying structure of the graph data largely determines the visual approach which can be taken. von Landesberger et al. [79] divide their discussion into trees (those with hierarchical structure), general graphs (which may be directed, undirected or mixed) and compound graphs (those with both hierarchical structure and other relations between nodes). The two main ways to represent general graphs are node link diagrams or matrix representations. Schulz and Schumann [80] distinguish three possible ways in which network visualisation techniques can be categorised:

- directed vs undirected
- explicit vs. implicit edge representation
- free, styled, or fixed node layout

Similarly, for tree representations, Schulz [123] identifies three 'design axes':

- dimensionality (2D, 3D, or hybrid)
- edge representation (explicit, implicit, or hybrid)
- and node alignment (radial, axis-parallel, or free)

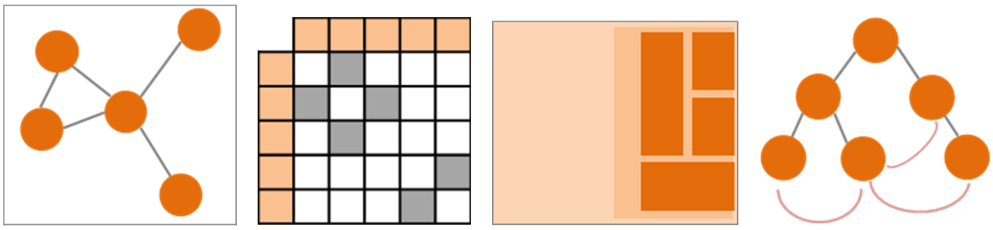


Figure 58 Possible graph structural encodings (left to right): node-link, matrix, space-filling, compound graph representations.

A simple classification is used in the design space, dividing the graph structural encoding dimension into the following general categories (illustrated in **Figure 58**):

- Space filling (enclosure, adjacency, overlap)
- Node-link
- Matrix
- Compound Graph representations
- Other (including no structural encoding e.g. topological statistics only)

For the sake of simplicity, directionality, dimensionality and node alignment are not used to classify the representations.

7.1.2.2 Temporal dimension

Considerable work has been carried out in visualising general time-oriented data [3], [98], [121]. Aigner et al. [98] distinguish the possibilities for visual representation by whether time is mapped to space (static) or time (dynamic), and the dimensionality of the presentation space (2D or 3D). However, the possibilities for temporal graph visualisation are restricted by the need to show both graph structure and time. Moody et al. [29] note that a key problem is that the two spatial dimensions - the most salient visual channels - are usually taken up in laying out the graph, raising the question of how to represent the temporal dimension.

In classifying the approaches, in addition to extracting those commonly discussed in the temporal graph literature, Javed and Elmqvist's [39] design patterns for composite visualisation (juxtaposition, superimposition, overloading, nesting, integration), and Gleicher et al.'s [102] categories of comparative designs (juxtaposition, superposition, explicit encoding), were drawn on. The following temporal encoding categories were identified (illustrated in Figure 59 and Figure 60):

(1) sequential views
(2) juxtaposition
(3) additional spatial dimension
(4) superimposition
(5) merged views
(6) nested views
(7) time as a node in the graph.

These categories can be grouped based on whether multiple temporal snapshots are presented (1-4), or time is 'embedded' within the graph structure (5-7).

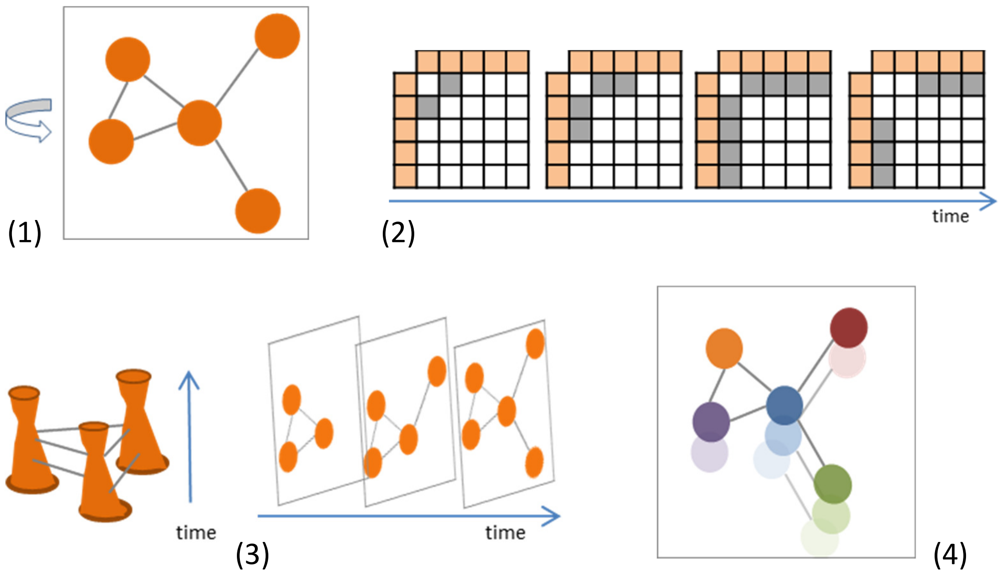


Figure 59 "Timeslice" approaches to temporal encoding: (1) sequential views, (2) juxtaposition, (3) additional spatial dimension, (4) superimposition.

The first four approaches show a series of what Archambault et al. [27] refer to as 'timeslices': snapshots encoding the structure of the graph at a given time. These approaches require particular consideration to be given to the readability and computation of the layout of the graph structure at each timeslice. Much work to date has focussed on the computational difficulties of adapting and developing layout algorithms for dynamic graphs [25], [26], [124]–[130], given the trade-off between the accepted set of aesthetic heuristics for (static) graph drawing and maintaining an individual's 'mental map' over a series of timeslices. Work has also been devoted to assessing the resulting representations in terms of user comprehension [16], [19]–[21], [131]–[133]. These 'timeslice approaches' can be divided based on whether the timeslices are mapped to time (dynamic presentation) or space (static presentation).

Sequential views are dynamic: timeslices are presented one after the other, in sequence, each replacing the last. Navigation through the timeseries may be automated (play/pause functionality) or interactive (e.g. through use of a timeslider). Transitioning techniques, such as animation and interpolation of node positions, may be employed to assist people in following changes between timeslices. Note that the literature often refers to these approaches as "animation", however, the term 'sequential view' was chosen in order to avoid ambiguity, as the term "animation" is used in two ways:

(1) *animated navigation*: where navigation through the sequence of timeslices is animated i.e. where the person using the system presses a play button and is shown an automated sequence of images, similar to playing a movie, and

(2) *animated transitions:* where animation is used to smoothly interpolate the positions of nodes between timeslices i.e. they do not just jump from one position to another, but their transitions are animated across the screen.

These two aspects often appear together, however it is useful to separate them out, particularly as what are referred to as 'animated' views in the literature often do not involve the animated navigation described in (1), rather, they allow interactive control of the navigation through timeslices.

The other three approaches are static. Examples of **juxtaposition** are most often akin to Tufte's [134] 'small multiples', with timeslices laid out adjacent to one another in sequence. However, in the design space, Gleicher et al.'s [102] wider definition is adopted, to include in this category examples where timeslices are positioned separately, but in the same display space. For example, TimeRadarTrees [135] and Tree-ring Layouts [136] use concentric circles to indicate the temporal aspect of the network. Also included in this category are general time series views of graph-based statistics (where statistical values represent the graph or its attributes at multiple points in time), and alluvial diagrams [137], which plot node-related statistics (topological or attribute based) as lines over time, with relatedness in the graph represented by positioning the nodes' timelines closer together.

Where an **additional spatial dimension** is used, timeslices are either presented as separate layers on an additional plane (e.g. Federico et al.'s '2.5D' approach [28]) or the nodes of the timeslices are 'stacked' resulting in three dimensional objects e.g. [138], [139]. **Superimposition** [39](also termed 'superposition' by Gleicher et al. [102]) involves overlaying objects in the same display space. In the temporal graph case, timeslices are stacked on top of one another and 'flattened', with a visual variable (such as colour, transparency, etc.) distinguishing elements belonging to different timeslices [28]. This results in the same nodes and edges appearing more than once in the same view.

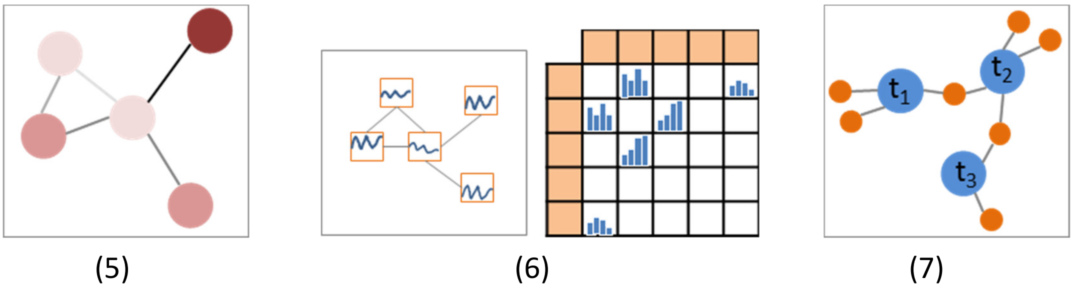


Figure 60 "Embedded" approaches to encoding the temporal dimension: (5) merged, (6) nested, (7) time as a node in the graph

Approaches 5-7 embed the temporal dimension within a single graph structure. **Merged views** differ from superimposition in that they show a single (cumulated) graph structure (i.e. each node appears only once), and use an additional encoding (e.g. colour) to indicate ageing of nodes and edges. **Nested views** [39] in the temporal graph case show the temporal aspect of the data by embedding small timeseries charts or glyphs in the nodes and/or edges. A bipartite graph including **time as a node** can be created; any node linked to a time node indicates that it appeared in the graph at that time. A variation of this is 1.5D [140], where a focus node contains an embedded timeline glyph and other nodes connect to the appropriate section of the timeline.

Note that Javed and Elmqvist's [39] integration category, which involves the use of visual links between views, and Gleicher et al.'s [102] explicit encoding, where the relationship between two objects is computed and visually encoded, are not included as categories in the design space, which is concerned solely with temporal and graph structural encodings. However, these techniques may be used in conjunction with the timeslice approaches of the design space to encode relations i.e. to show the differences or matches between timeslices. This is often of interest in temporal graph visualisation, which is closely related to graph comparison, and is discussed further in Chapter 8. Visual links are often used in conjunction with 2.5D views to map node positions between timeslices e.g. [28], [141]–[143], but could potentially be used with any of the static timeslice approaches (i.e. approaches 2-4). There are many examples of explicit encoding in the graph comparison literature: difference maps [144], difference layers [145], ratio contrast treemaps [146]. Used in conjunction with a timeslice approach e.g. [141], they can show the evolving relationships between timeslices over multiple different time points. Finally, in Javed and Elmqvist's [39] overloading category the space of one visualisation is utilised for another. Some examples of this can be seen when views are combined, as discussed in Chapter 8.

7.2 Structure of the design space

Based on the two identified dimensions, a matrix has been constructed which maps out the possible combinations of graph and time encodings (Figure 61).

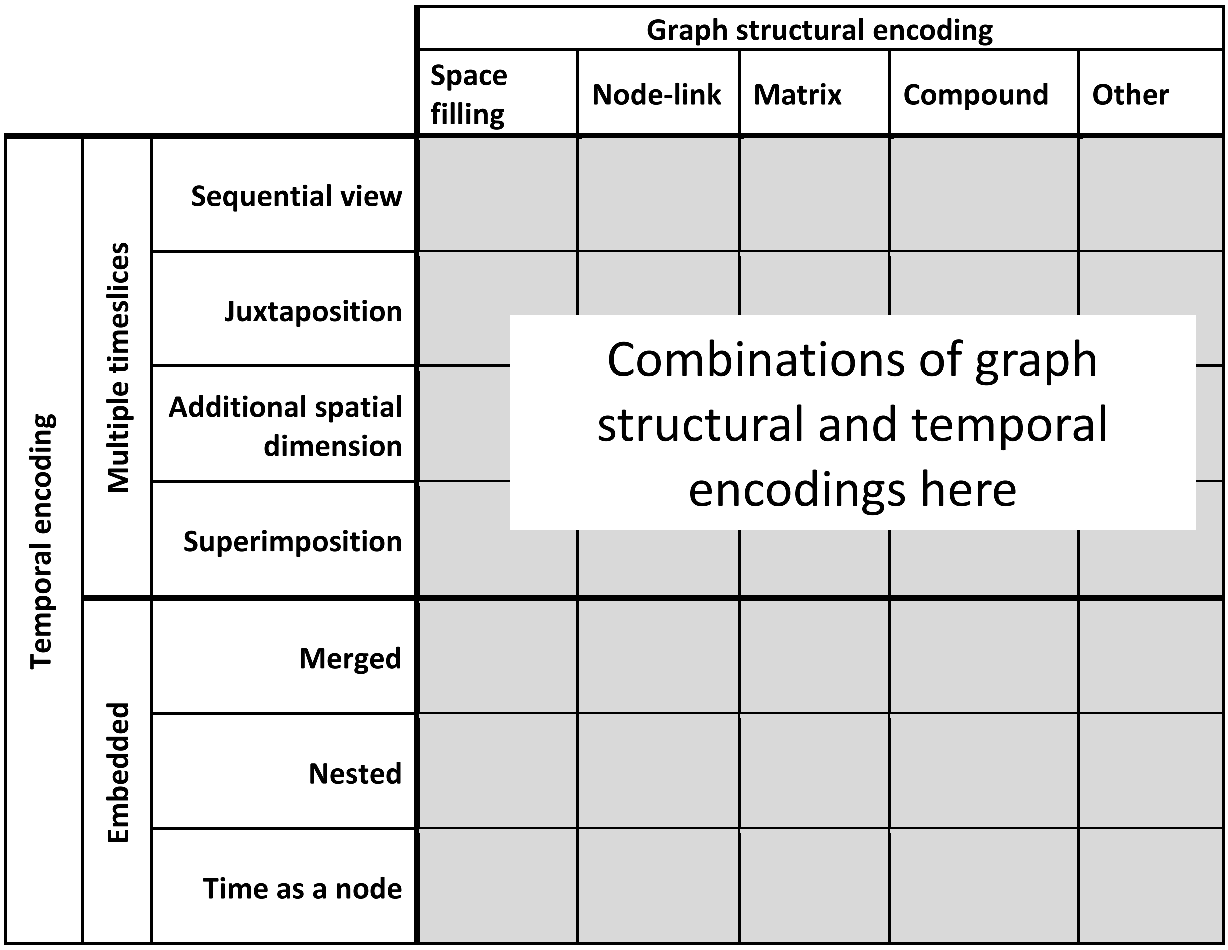


Figure 61 Design space of temporal graph visualisation techniques

7.3 Mapping Existing Techniques to the Design Space

Having constructed the design space, existing techniques have been mapped to each of the cells. 95 papers relating to temporal graph visualisation have been surveyed, including system and technique papers, comparative evaluations of techniques, and those discussing the use of tools to perform analysis. The combinations of encodings that were used in these papers were mapped to the appropriate cells in the design space. Where a paper discussed multiple techniques, it was included in all of the relevant cells. In total, 128 instances of techniques were mapped to the design space.

Note that Ahn et al. [41] include a list of online materials in their review of systems, however these were omitted as the vast majority are of the node link, sequential, category. Examples from the mapping are included in Figure 62; the complete mapping is shown in Figure 63.

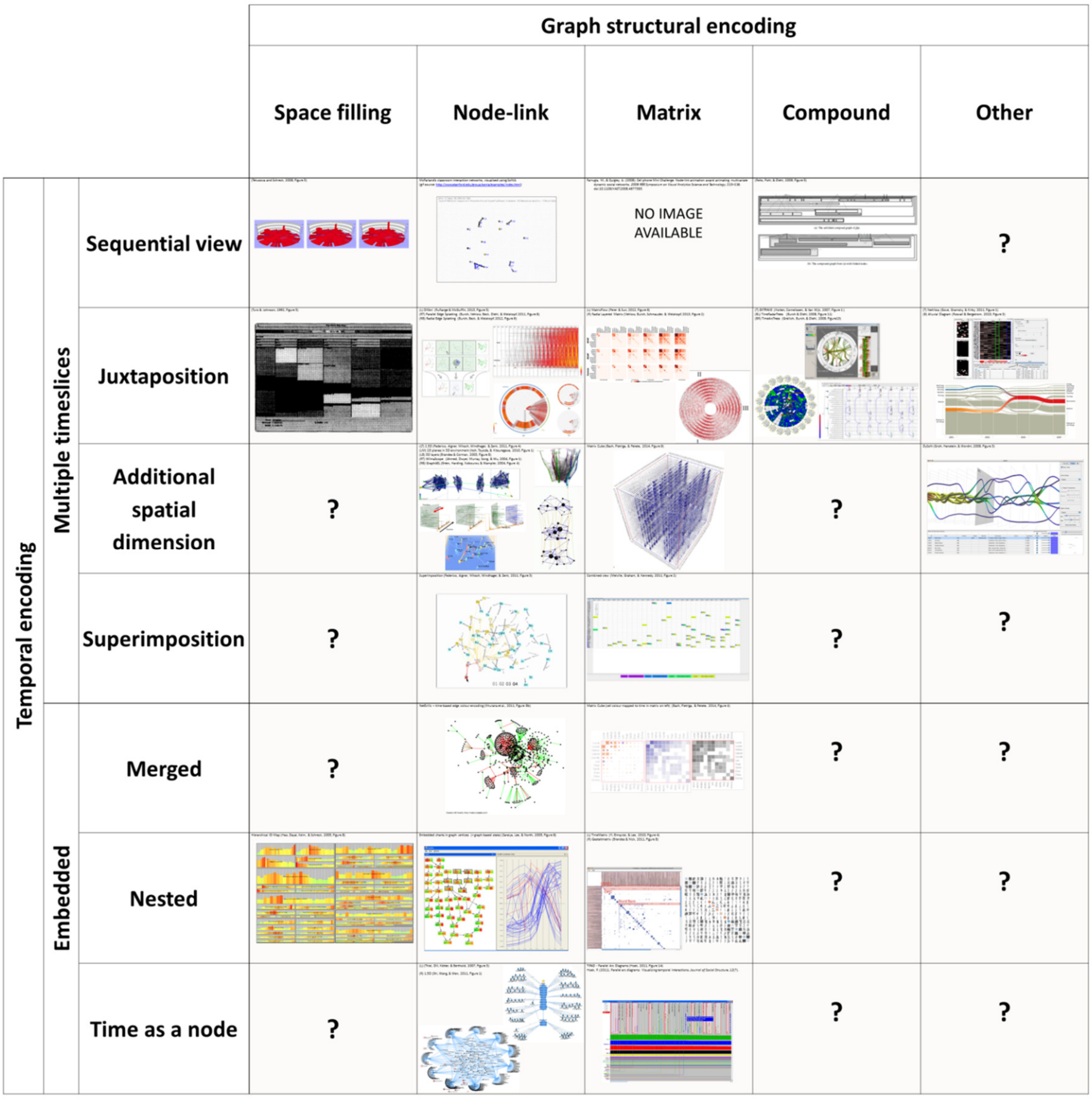


Figure 62 Mapping of techniques to the design space: an example image is included to illustrate where there exists a mapping to the itereture

			Graph structural encoding				
			Space filling	Node-link	Matrix	Compound	Other
Temporal encoding	Multiple timeslices	Sequential view	[147] [148] [149] [150] [151]	[152] [153] [154] [155] [23] [156] [157] [158] [29] [159] [160] [75] [161] [106] [76] [162] [163] [164] [165] [25] [166] [125] [167] [168] [169] [170] [171] [172] [78] [173] [174] [27] [127] [175] [124] [126] [176] [177] [178] [179] [180] [181] [182] [183] [184] [83] [71]	[165]	[185] [24] [186] [187]	
		Juxtaposition	[148]	[23] [188] [167] [189] [78] [173] [27] [190] [28] [136] [177] [191] [178] [179] [192] [182] [83] [193] [194] [71]	[195] [196] [197]	[186] [135] [198] [112]	*Graph statistics* [199] [200] [201] [104] *Alluvial diagrams* [202] [137][203]
		Additional spatial dimension		[204] [139] [138] [23] [142] [143] [199] [173] [28] [182] [192] [194]	[197]		[205]
		Superimposition		[173] [28] [192] [182] [194]	[195]		
	Embedded	Merged		[206] [172] [126] [207] [140]	[197]		
		Nested	[208] [209]	[73] [112]	[210] [70] [211]		
		Time as a node		[212] [140]	[213]		

Figure 63 Mapping of existing techniques in the literature to the design space

7.3.1 Findings

All 128 techniques were mapped to the design space, indicating that the categorisations used are appropriate. The number of techniques mapping to each category are shown in Figure 64.

			Graph structural encoding				
			Space filling (8)	Node-link (93)	Matrix (11)	Compound (8)	Other (8)
Temporal encoding	Multiple timeslices	Sequential view (57)	5	47	1	4	
		Juxtaposition (35)	1	20	3	4	7
		Additional spatial dimension (14)		12	1		1
		Superimposition (6)		5	1		
	Embedded	Merged (6)		5	1		
		Nested (7)	2	2	3		
		Time as a node (3)		2	1		

Figure 64 Mapping of techniques to design space

The most common graph structural representation encountered in the temporal graph visualisation literature was node-link. This is in-keeping with findings from the static graph literature, where the majority of systems are node-link based [214]. Matrixes are particularly useful for visualising dense networks due to the absence of edge crossings, and they have been shown to outperform node-link diagrams on a number of tasks in the static context [215]. Further research could therefore be applied in this area. There is also room for further exploration of temporal visualisations utilising space filling techniques.

While a number of examples of juxtaposition were found, sequential views were by far the most widely used temporal encoding. This is interesting, as juxtaposed views have performed well in a number of studies comparing them with sequential approaches [18], [78], [177]. The other approaches to temporal encoding featured less prominently in the literature.

There are a number of gaps and sparsely populated cells in the design space. While there may be good reason for this (e.g. incorporating time as a node in a space-filling representation would not be possible given that a hierarchical graph structure is required), the mapping shows some possible interesting directions for further exploration.

7.4 Strengths and weaknesses of temporal and graph encodings

Having considered the range of possible time and graph encodings, and their use in existing systems, let us now consider in more detail their relative strengths and weaknesses. This discussion is divided into two parts according to the two dimensions of the design space: graph structural encoding and temporal encoding.

7.4.1 Strengths and weaknesses of graph structural encodings

Each of the techniques for encoding graph structure has relative advantages and disadvantages. When selecting a representation, our choice of graph encoding is likely to be influenced by the type of graph structure present in the data. General graphs can be represented using node-links or matrices; combined versions of these have also been used in static graphs e.g. [214], [216], although they are yet to be used in a temporal graph context. Hierarchical structures have the additional option of space filling techniques (similarly, combined views, utilising node-link and space filling techniques have also been proposed in the static case e.g. [217], [218]). Compound graphs require two representations of graph structure: one to show hierarchy and another to show the additional links within the hierarchy, and various combinations have been used in the literature (**Figure 65** shows examples from Holten [219]).

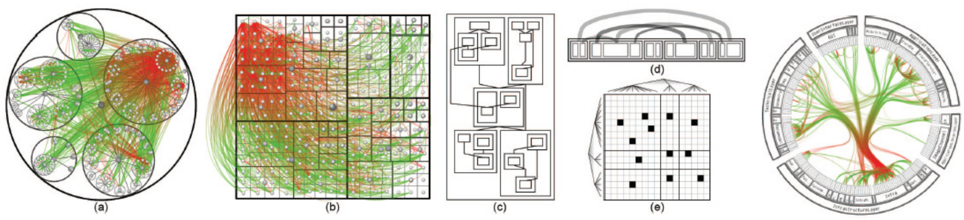


Figure 65 possible techniques for representing compound graph structures (illustrations from Holten [219]**, Figures 2 and 13b): (a) node link hierarchy + node link (b) & (c) space filling hierarchy + node link (d) space filling hierarchy + node link (arc diagram) + (e) node link hierarchy + matrix (f) space filling hierarchy + node link (with edge bundling)**

Node-link diagrams are a commonly used representation which are intuitively understood by people, however they do not scale well. As the size of the graph increases, it becomes computationally more expensive to calculate the position of nodes and takes longer to render. Readability also suffers, with nodes and edges overlapping, and occlusion making interaction difficult; eventually we simply run out of screen space in which to draw nodes and edges.

With no edge crossings, matrices do not suffer from the same readability issues and computational overheads associated with laying out large graphs. However, matrices are less intuitively understood than node-link diagrams, and node-ordering algorithms are required to show clusters.

Ghoniem et al. [215] compared the performance of node-link diagrams and matrices to carry out seven commonly encountered graph-related tasks:

- estimating the number of nodes or edges in the graph
- finding the most connected node, or a node given its label
- finding a link, a common neighbour, or a path between two specified nodes.

They found that on graphs of size greater than 20 nodes, matrices outperformed node-link diagrams on all tasks except path following in terms of speed and accuracy in participant performance. Keller et al.'s [220] comparison of node-link and matrix representations used semantic, directed graph data and slightly different tasks, but their findings that node-link graphs are preferable for small, sparse, graphs and for path finding tasks, confirmed those of Ghoniem et al. Note, however, that neither of these studies considered tasks involving clusters (Keller et al. suggest these should be investigated in future work) or graph attributes.

No study has yet compared the performance of matrices and node-links in representing clusters. Siirtola & Mäkinen's [221] study found that the use of an automated re-ordering algorithm for matrices allowed participants to perform cluster analysis with more accuracy and in less time than using a static matrix, or one with manual reordering capability. However, they did not compare the performance of the re-ordered matrix to a node link diagram. Wong et al. [222] demonstrate some examples of how their approach - which combines a pairwise shortest distance matrix with a node-ordering algorithm - is better able to show clustering patterns than a node-link diagram. However they did not carry out a systematic evaluation comparing the two approaches.

In addition, no studies exist which consider the performance of node-links and matrices with regard to their ability to represent distributions of graph attributes. Matrices perhaps have an advantage over node links of more encoding possibilities for edge weights/ attribute values (see Section 8.2.2.1). Understanding node attribute distributions over a graph usually requires a node-link view of the data; whether distributions of edge weights/attributes are perceptible using matrices has not been studied.

As technique performance has been shown to be data and task dependent, Keller et al. [220] recommend offering multiple views i.e. both matrix and node links, so that people can choose the most appropriate representation for the task they are attempting to carry out. Alternatively, approaches combining matrix and node-link structures (e.g. [214], [216]) seek to draw on their respective advantages. NodeTrix [214] was developed to visualise small world networks, which are characterised by tightly connected clusters with sparse links between them. Matrices, representing tightly connected clusters (dense graph structures), are linked together using node-links, showing the sparse, global structure of the network. Such combined graph encodings have yet to be used in a temporal graph scenario.

For hierarchical structures (**Figure 66**), matrix views are not normally employed due to the difficulties in path following and their being space-inefficient for this type of data [81]. Generally node-link representations take up more screen space than nested space-filing techniques such as tree maps, but hierarchical structure is more difficult to perceive in space-filling representations, which also emphasise leaf nodes over internal nodes [81]. Adjacency layouts (such as icicle plots), which are a type of space-filling layout, trade off the advantages and disadvantages of node-links and tree maps. Edge attributes cannot be encoded when using space-filling techniques (see Section 8.2.2.1).

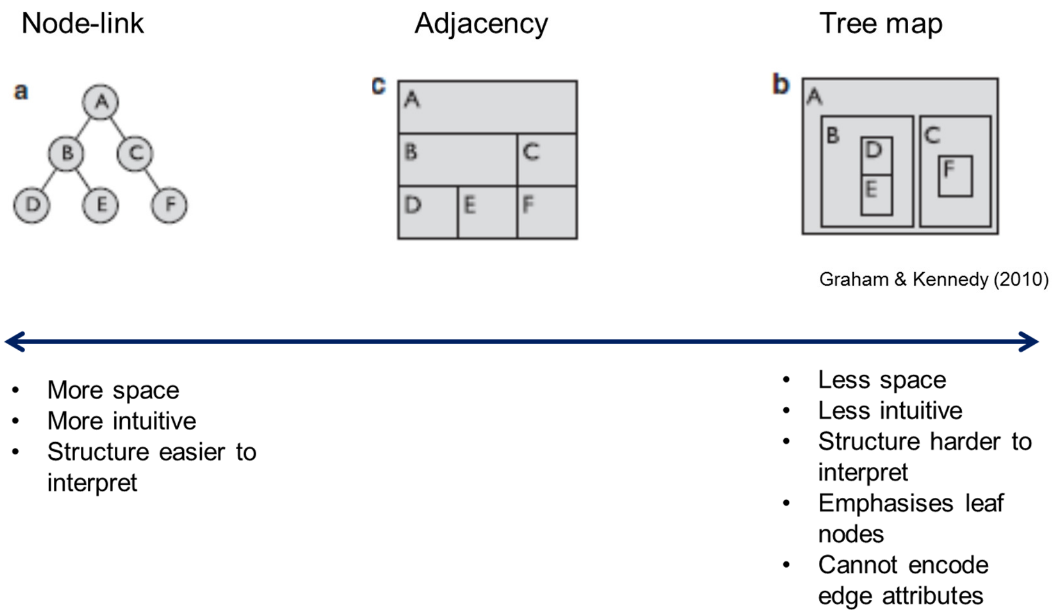


Figure 66 comparison of node link and space filling techniques for hierarchical graph structures (based on Graham & Kennedy [81]**)**

Müller et al. [223] recently compared hierarchical visualisation techniques in terms of their ability to "*facilitate a rapid overview of the structure and intuitive impression of proportions between nodes*". They considered the three most popular top-down techniques (node-link, icicle plot, and squarified treemap) and four tasks (their favoured representation for each technique is in brackets):

- *Count all nodes of the hierarchy. (node-link)*
- *Count leaf nodes of the hierarchy. (treemap)*
- *Compare the combined area of two pairs of nodes within one level of the hierarchy. (icicle plot)*
- *Compare the combined area of two pairs of nodes across different levels of the hierarchy. (equal)*

Assuming that area represents an attribute, this study did consider attributes in their tasks (unlike the node-link/matrix comparison studies), however, they did not consider attribute distribution over the graph's structure. Again, further work in this area is required before a recommendation can be given as to which technique is best able to represent overall structural patterns and attribute distributions.

7.4.2 Strengths and weaknesses of temporal encodings

Seven distinct encoding approaches were identified along the temporal dimension of the design space (Section 7.1.2.2). As for graph encodings, each approach has relative strengths and weaknesses, which are discussed in this section. Note that to date, empirical evaluations involving participants which specifically compare temporal encodings (e.g. [27], [78], [177]) have focussed solely on comparing sequential views and juxtaposition (small multiple views). Additionally, all of the studies used node-link graph encodings and the size of the graphs and number of time points involved are relatively small: Archambault's comparison of animation and small multiple conditions [27] employed graphs with 29-60 nodes and nine time points; Farrugia and Quigley's [78] graphs contained between 9 and 32 nodes, and six time points; Boyandin et al.'s qualitative study [177] used the largest data sets, with graphs of around 200 nodes and 35 time points. The consideration of attributes in these studies is also rather limited: edge weights are considered in Boyandin et al.'s flow maps, and Farrugia and Quigley use colour and shape to encode attributes, however attributes do not feature in their example tasks or discussion of results. One final issue with these studies is that limited interaction is offered in the sequential view conditions, which may have curtailed the potential benefits of using a sequential approach [78]. The results of these studies are generally rather inconclusive, other than indicating that for tasks involving more than two time steps, small multiple encoding may be preferable [84].

Let us consider in more detail the merits and drawbacks of each of the approaches, beginning with sequential approaches, where time is mapped to time.

7.4.2.1 Sequential approaches

The advantages of sequential approaches include:

- Time is encoded in a 'natural' way, instinctively understood by people.

- Time is an encoding channel which cannot really be used to encode any other attribute; this ‘frees up’ an additional visual variable when encoding multidimensional data.
- The full display space is available to show the graph structure at a single time point, which is particularly important as graphs become larger and clutter and occlusion become more of an issue.
- In terms of tasks from the empirical studies, Archambault et al. [27] found that animation was more accurate for a minority of tasks (those relating to the addition of entities), while in Boyandin et al.’s [177] study involving the exploration of flow maps, participants made more findings involving “*geographically local events and changes between subsequent years*” under the animation condition.
- Animated transitioning techniques between timeslices can be used as an additional technique to draw attention to change in the graph. Windhager et al. [224] highlight this as an advantage, in that it can enhance perception of change between time slices and reduces change blindness. However, such interpolation also introduces additional artifacts which do not exist in the data [78].
- In trend visualisation, Robertson et al. [225] noted the usefulness of animation in presenting information.
- In the same study, they note that participants found the animation paradigm enjoyable and “exciting”; however, the participants in Farrugia and Quigley’s [78] study comparing animated and small multiple approaches in the temporal graph case showed a preference for the static condition.

The disadvantages of sequential encoding include:

- A key disadvantage of sequential views is the cognitive overhead involved. The person using the visualisation must memorise what has taken place in the graph as the timeseries unfolds; even comparison of adjacent timeslices must be performed in memory. While transitioning techniques and differencing techniques can help mitigate these problems between individual timeslices, without some additional visual support (such as the thumbnails of GraphDiaries [71]) when exploring and analysing temporal graphs, navigation relies entirely on a person's memory of the sequence of events.

- The above is perhaps one reason that evaluative studies find that tasks take longer under animation conditions, as participants need to view the whole sequence of graphs before being able to answer the prescribed questions.
- In animated navigation (i.e. 'play only' scenarios, which unfold in a film-like manner) the lack of interaction makes it difficult for people to explore the data.

7.4.2.2 Other timeslice approaches

The advantages of the other multiple timeslice views - juxtaposition, additional spatial dimension, superimposition – include:

- The data is available at once, in a single display space (depending on the length of time-series/size of the graph – see discussion, below). This removes the issues surrounding the cognitive overhead associated with animated displays, where memorisation of the occurrence of events at previous time points is required for analysis of global change, and navigating the timeseries.
- Evaluative studies (such as Archambault et al. [27] and Farrugia and Quigley [78]) suggest that small multiple approaches are generally faster and more accurate when performing most tasks.
- Farrugia and Quigley [78] found a preference among participants for their small multiples condition.
- Boyandin et al.'s [177] subjects were able to make more findings concerning longer time periods using small multiples of flow maps. This would lend support to using this type of temporal encoding when visualising Q4 behaviours.

The main disadvantage for these encodings come from the mapping of time to space and involve scaling of the data, either the graph component or the temporal component:

- Limits to the space available to show each individual timeslice, making it difficult to show large networks and/or details (e.g. attribute encodings, labels, etc.).
- Limits to the number of timeslices which can be shown (see Section 7.4.2.3).

7.4.2.3 Timeslice approaches: problems of scale in the temporal dimension

One fundamental issue for all of the timeslice approaches is how to handle scale on the temporal data dimension. In the case of animated approaches, where time is mapped to time, longer timeseries result in longer animations. As evaluative studies (such as [27]) suggest that animation is slower than static approaches for most tasks, one could infer that this difference will become more pronounced with an increasing number of timeslices (unless some form of temporal aggregation is used, such as that discussed by Bender-deMoll and McFarland [159]).

For static approaches, with a spatial mapping of the temporal data dimension, the issue is one of available space. In juxtaposed views, assuming a limited total display space (such as a printed page or computer screen), an increasing number of timeslices reduces the individual display space available, thus reducing the amount of detail and/or legibility of the graph in each slice. In superimposition, where timeslices are layered on top of one another with a visual variable distinguishing different time steps, visual clutter increases with the number of timeslices (this is likely to become a problem before a second issue - limits to discriminability in the visual variable distinguishing the time steps – is reached). Where an additional spatial dimension is used, if the total display space is fixed, issues of occlusion/distortion could be introduced as more layers are compressed into the display space. In addition, for this paradigm to be understandable to the viewer, there is perhaps an increased requirement for stability in the layout between adjacent layers, and indeed, over the time period. For example Brandes and Corman's [139] cylinders require a fixed layout (although Groh et al.'s [205] inter-twinning tube solution is a counter example to this). In both juxtaposed and superimposed views, the limits on display space can be removed by showing only part of the timeseries and allowing some sort of interactive navigation (such as scrolling). However, this reduces the advantage of being able to see all of the data at once.

7.4.2.4 Embedded approaches

So far, only encodings employing multiple timeslices have been considered. Let us now discuss the merits and drawbacks of embedded temporal encodings: merged views, nested views, and time as a node in the graph.

As mentioned earlier, nested views show behaviours S3 and A3 (i.e. distribution of temporal trends over the graph). However, they are also subject to limitations in the display space available to show the temporal dimension. In this case, consideration must be given to the length of the timeseries/number of legible timepoints that can be displayed. Limits to the size of the graph which can be shown must also be taken into account. Yi et al. [70] use semantic zooming techniques in order to display graphs with many nodes (730 in their example dataset) and edges.

Merged views use a visual encoding such as colour or intensity to indicate ageing and/or persistence of nodes or edges in a network. They are most often used in conjunction with another temporal encoding (such as sequential views), as alone, merged views can show only limited aspects of a graph's evolution. They have the advantage of using the full screen space to lay out the graph, and show an overview of certain features of the data. For example, they can give an overview of the formation of a network. A network which gradually grew over time could be indicated by an even distribution of the visual encoding used; where lots of nodes joined at the same time would be indicated by similar visual encoding; while some particular distribution (old nodes in the centre, new ones at the periphery, etc.) may also be observed. The main limitation of merged views is that they are able to show only limited aspects of the data i.e. either structural additions or deletions, otherwise the visualisation becomes too difficult to understand. For this reason, Smuc et al. [226] abandoned the development of their SPOCC plots which use colour to encode addition, deletion and persistence of edges between two timeslices. Showing a node which appears at time 1, disappears at time 2, and re-appears at time 3 cannot be captured in merged views. Merged views may also suffer from problems of clutter and occlusion, as they show all nodes and all edges appearing in the graph. Additionally, there are limits to the number of time points which can be distinguished accurately through use of a visual encoding such as colour or intensity, and the use of a visual variable to encode time means one fewer encoding channel is available to encode attributes.

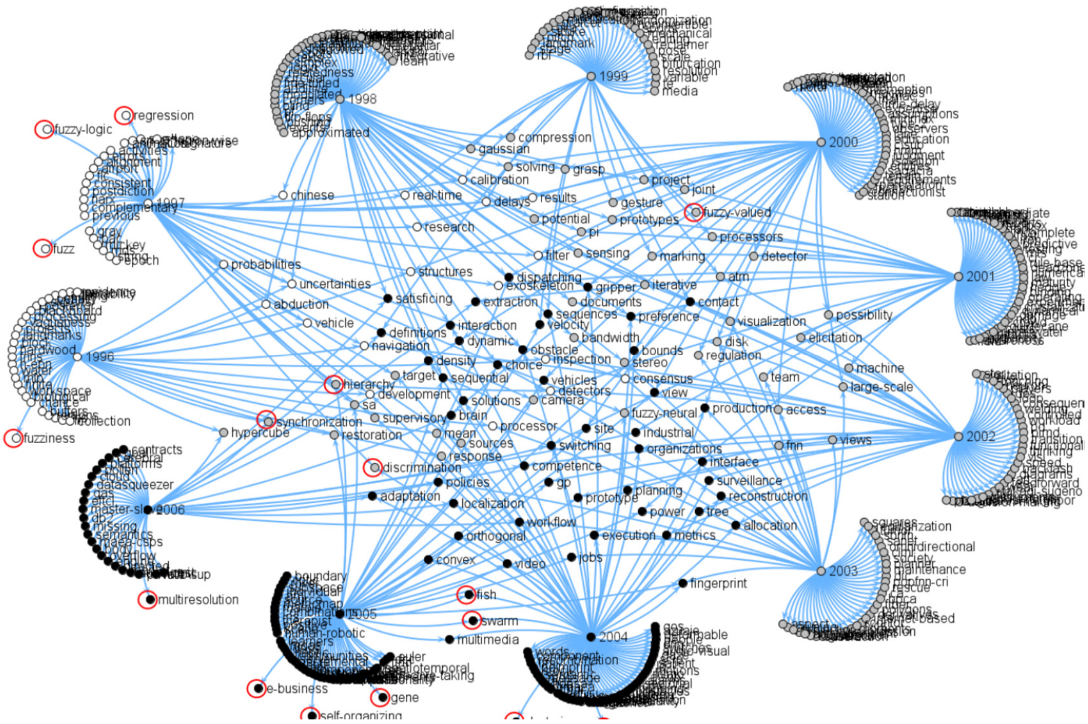


Figure 67 Thiel et al.'s ([212]**, Figure 3) bipartite graph shows the years in which a node (in this case representing a keyword) appears: the edges encode a node's appearance in a given year.**

In its basic form, representing time as a node in a graph (e.g. [212], illustrated in **Figure 67**) primarily allows us to gain an overview of structural patterns in node activity over time i.e. which nodes are persistent in the network, and which nodes appear only at individual time points. However, it does not give any indication of evolution in structural patterns, or attribute distributions over time. The technique has the advantage of showing all of the data in a single display space (much like the timeslice approaches). However, all nodes (plus year nodes) are shown for the whole time period, which may be an issue for large graphs, and the size and/or density of the graph increases with the number of time points.

7.4.3 Summary

As the above discussion has shown, scaling along the time and graph dimensions are issues for many of the encodings. Reduction and filtering techniques can be helpful in reducing these problems, and may also be useful when finding patterns at different granularities. For example, when nodes are aggregated in TimeMatrix [70], aggregated timeseries data is displayed for both nodes and edges, while Shi et al.'s 1.5D [140] display allows interactive selection of focal nodes, and different levels of temporal granularities.

The above discussion also indicates that more work is needed in evaluating the different encodings in order to indicate the scenarios in which they are most effective. For example, with the present information, it is not possible to make recommendations as to which technique is best for long timeseries, highly volatile data (i.e. where there are large amounts of change in the graph) or where change is highly irregular (e.g. long periods of no change, followed by many changes). Further work is also needed to evaluate how the techniques are able to support the analysis of attribute values, for example, to establish which technique is best able to show evolution of attribute distributions.

7.5 **Summary**

This chapter has considered the existing visual approaches for temporal graph data and explored the 'space of the possible'. A design space was constructed according to the temporal and graph structural encodings used, to which the existing techniques were mapped. The mapping showed the most commonly used techniques, and possible combinations of encodings which could be further explored.

The discussion relating to the strengths and weaknesses of the possible visual encodings highlights the need for further empirical evaluations. For example, the performance of different graph encodings (matrices and node-links) in terms of their support for representing clusters and attribute values has not yet been established; comparison of techniques for visualising the evolution of distributions of attribute values could be carried out; and further work is needed in comparing temporal encodings in order to establish which techniques are most suitable for representing temporal evolution, and in which data scenarios (e.g. large graphs, long time-series, volatile data etc.).

Chapter 8 Mapping tasks to visual techniques

In this chapter, the visual techniques which are able to support the tasks of the taxonomy are considered. Given that a single task may be categorised in a number of ways, and the sheer number of individual tasks identified in the design space, the discussion is structured around the task dimensions. Primarily considered are task type and quadrants, but also considered are the implications of additional constraints in lookup tasks (search space), and, in the case of comparison and relation seeking, the involvement of a specified component or the same or two different graph components, time components, or attributes.

This chapter is organised as follows. The role of the quadrants in determining the appropriateness of visual techniques is first discussed. Next, each task type and techniques for their support are considered, with reference to the quadrants. A discussion of the implications of task search space is included in the inverse lookup task section. Techniques to handle the variations of the same or different time, graph, or attribute components participating in tasks are discussed at the end of the comparison section. As each task is likely to be only one of many involved in exploratory analysis of data, the ways in which techniques can be combined are considered. Finally, when mapping techniques to tasks, a number of areas which could benefit from further research are identified; we conclude with a discussion of this in relation to the findings of the evaluation of the task framework in Chapter 6.

An overview of the task-technique mapping is included for reference in Section 8.5.

8.1 The role of the quadrants in determining appropriate visual techniques

As briefly mentioned in Section 5.5, visually representing the data items in each of the four quadrants involve very different techniques and research areas, as illustrated in Figure 68.

- Q1 (data elements and their attributes) is governed by general visualisation principles.
- Q2 is dealt with by static graph visualisation.
- Q3 is the domain of temporal visualisation.
- Q4 is the only quadrant requiring the representation of both time and graph structure, and therefore temporal graph visualisation techniques (such as those reviewed in Chapter 7) are involved.

However, *any* of these data items and associated techniques may feature in the exploration of temporal graph data. Within each category, decisions as to the appropriateness of a visual representation will depend on characteristics of the specific dataset. For example, when selecting a technique to encode graph structure (Q2), the size and density of the graph must be taken into consideration; when showing structural change over time (Q4), the rate of change and length of timeseries may influence our choice of representation (these considerations are discussed further in Section 8.2.2.4).

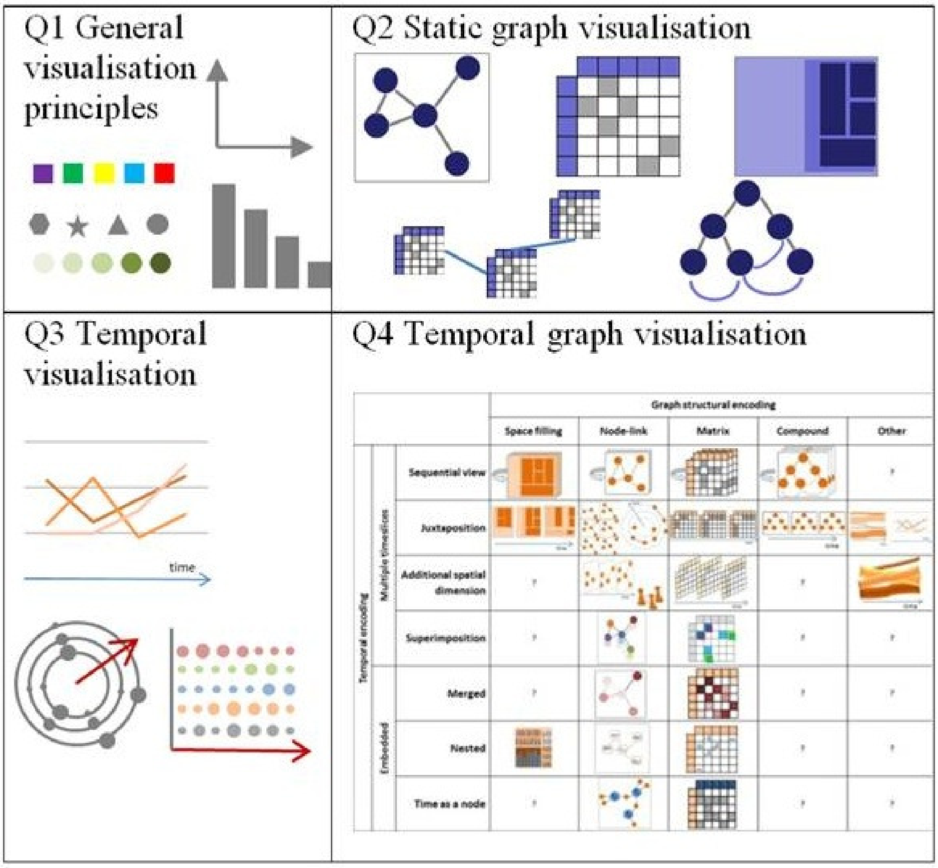


Figure 68 Research areas and techniques associated with data items by quadrant

8.2 **Lookup**

Direct and inverse lookup tasks require different techniques for their support as they take a different starting point for the analysis. The distinction is reflective of the bottom up ("*search, show context, expand on demand*" [227]) and top-down ("*overview first, zoom and filter, then details on demand*" [56]) information seeking approaches discussed more widely in the literature.

8.2.1 Direct lookup and behaviour characterisation

For direct lookup and behaviour characterisation tasks we must first locate the time and graph object of interest, in order to find the corresponding values and patterns. This requires navigation in both time and in the graph. Systems employing sequential views offer temporal navigation via interactive controls such as time-sliders e.g. TempoVis [228]; play/pause/step buttons e.g. SoNIA [159], Visone [125], Republic of Letters [171]; or thumbnails e.g. GraphDiaries [71]. Often a timeline of statistics relating to the network is shown in conjunction with navigational controls, which helps to summarise changes in the graph and draw attention to key time periods of interest. For example, TempoVis includes a histogram summarising node and edge activities in the graph over time (Figure 69), while Chang et al. [171] display the total number of edges at each time point on their scatter-line graph (Figure 70). Chang et al. also allow selection of a time period of interest over which to observe the animation of the graph, while TimeMatrix [70] includes a range slider to select a time period upon which to filter the matrix-based timeseries glyphs. Such techniques are particularly useful to support direct lookup tasks in Q4.

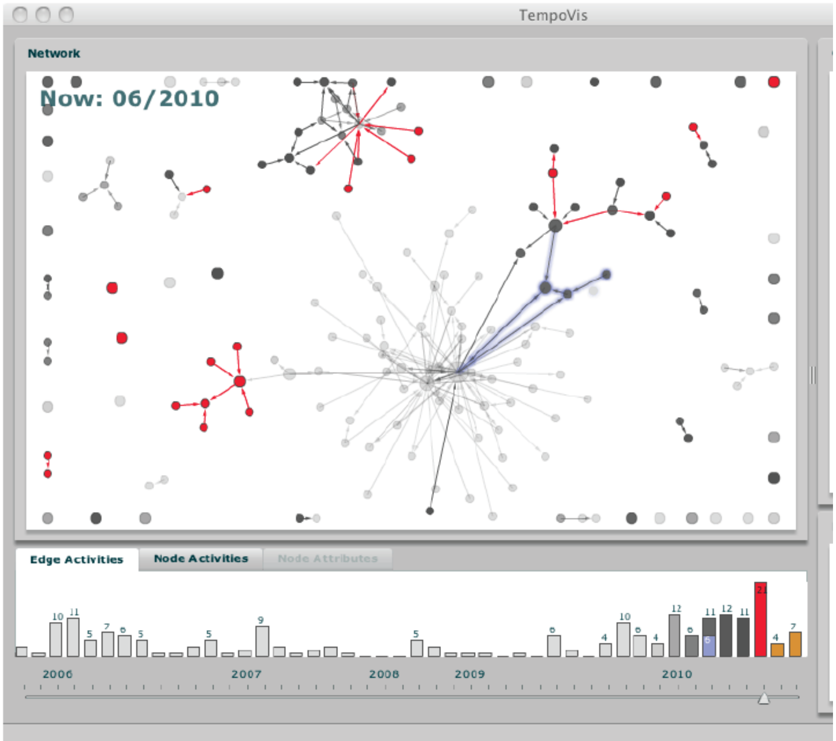


Figure 69 Ahn et al.'s TempoVis interface ([228], Figure 2) includes a time-slider for temporal navigation and histogram summarising activities in the graph over time. A time stamp (top left) indicates the current timeslice in the main window,

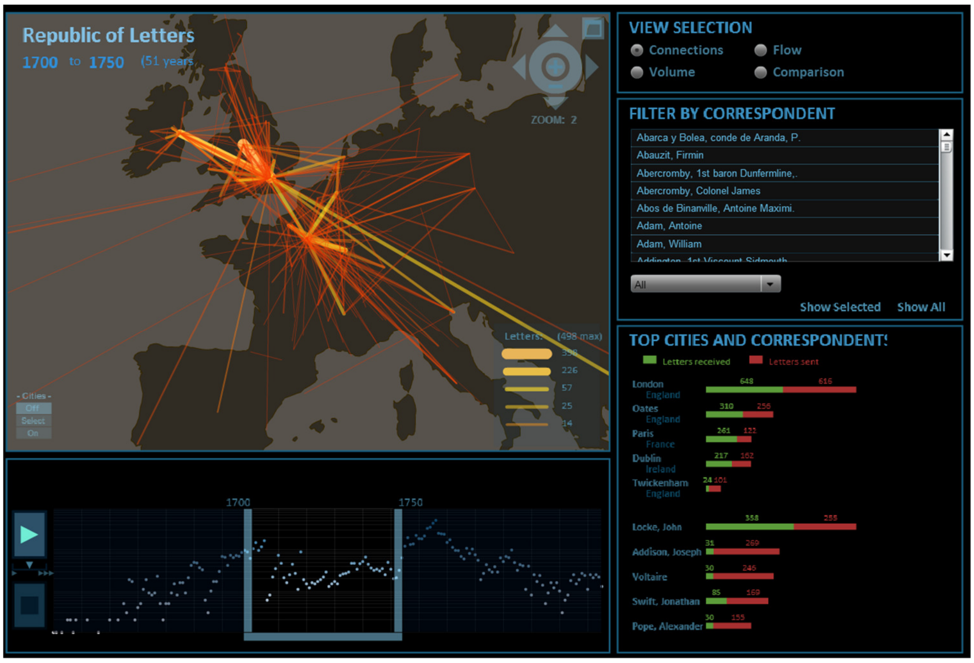


Figure 70 screenshot of Chang et al.'s [171] Republic of Letters visualisation (http://web.stanford.edu/group/toolingup/rplviz/), which includes play/pause temporal navigation controls, and the ability to select a time interval over which to play the animation. The scatter-line graph shows the amount of correspondence at each time point.

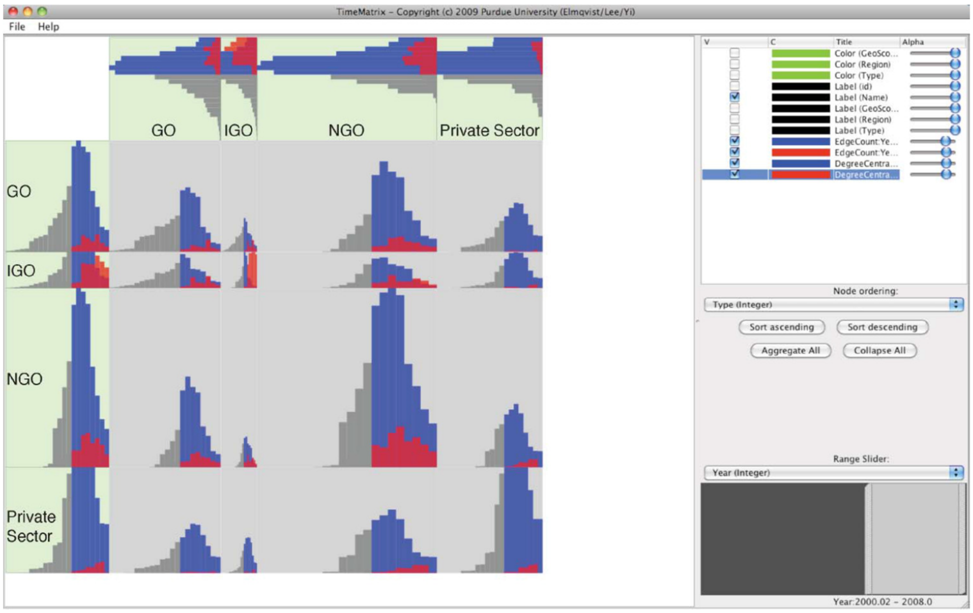


Figure 71 TimeMatrix ([70], Figure 6) offers filtering on time: time points out with the selected time range (as selected using the range slider, bottom left) are shown greyed-out in each timeseries glyph.

Locating particular graph elements of interest is potentially a more challenging tasks than locating time points, as graph elements (in a node link interface, at least) have no inherent order. As on-demand labelling strategies (discussed further in Section 8.2.2.6) are often employed, 'manually' finding a specific node may prove time consuming in larger graphs. To assist in this task, static graph systems, such as TaxVis [229], often offer a separate search box or list to filter and then highlight nodes of interest within the graph. An example from temporal graph systems is Chang et al. [171], who offer this functionality ("filter by correspondent" in Figure 70) and also highlight in their timeline view the time points at which the selected nodes appear. Interaction techniques such as pan and zoom can also be of use when locating graph elements in large graphs.

8.2.2 Inverse lookup and pattern search

Inverse lookup and pattern search tasks involve observing patterns and attribute values and identifying the corresponding graph objects and times of occurrence. As noted above, the patterns and values which we may observe are very different depending on the data items concerned, as distinguished by the four quadrants, requiring very different visual techniques for their representation.

8.2.2.1 Q1

In Q1, we are looking for particular attribute values, and the encodings used must be sufficiently distinguishable to allow this. The techniques used to represent individual nodes and edges and their attributes depend upon the graph representation used – node link, matrix, space filling. Generally speaking, to date, static graph and temporal graph visualisation has largely been focussed on representing graph structures, with less consideration given to representing attributes associated with the graph.

In node link diagrams, numeric attributes associated with nodes are often encoded by size (area), colour saturation or density, while hue or shape are frequently used to encode categorical attributes. Multiple node attributes may be incorporated in glyphs. There are some issues to consider when selecting attribute encodings: proximity is often used to indicate closeness of connection, but using size of node to encode attribute value may make it more difficult to judge the distance between two nodes. Edge attributes in node link diagrams are frequently encoded using line width or density/saturation (numeric) and colour or pattern e.g. dashed lines (categorical). Issues surrounding the use of line width include a limit of around five distinguishable bins [54], while altering line width can affect our perception of line length, and therefore potentially affect our perception of how closely two nodes are related. Altering the saturation of links can make them difficult to perceive.

Matrices focus on showing the relations between nodes (i.e. they focus on connectivity), although node attributes can be encoded using e.g. colour or shape (size would not normally be appropriate, as it may affect the width of the columns and rows in the matrix). The options for encoding edge attributes are more varied than in the node-link case, including colour, density, saturation, shape, and size of shape.

Space filling representations primarily encode attribute values using size, but an additional encoding such as hue or saturation can also be used. Position (enclosure or adjacency) represents edges, therefore it is not possible to encode edge attributes using this type of graph representation.

Dynamic query filtering techniques [230] can help find graph objects associated with particular attribute values e.g. SocialAction [161] offers filtering on node attribute values in static graphs, while Burch et al. [190], offer filtering on edge weights in their temporal graph system.

8.2.2.2 Q2

The timeslice views (sequential, juxtaposed, additional spatial dimension, superimposition) of the design space (Chapter 7) show a snapshot of the graph at an individual point in time i.e. a Q2 representation (partial behaviours S2 and A2).

In Q2, finding structural patterns depends on the graph representation used (discussed further in Section 8.2.2.4). Where a node link representation is employed, finding patterns is supported by the choice of layout algorithm; when using a matrix view, a clustering algorithm needs to be applied. Where node or edge attributes are visually encoded, the graph layout also determines how attribute distributions are perceived. DGD-Tool [167] offers a choice of layout paradigms for node link diagrams, depending on the patterns of interest to the analyst: force directed, which highlights clusters; layer based, for detecting hierarchic structure; and orthogonal, for detection of paths between connected nodes. Similarly, TVN Viewer [175] offers a choice of radial and force directed layouts. As discussed in Chapter 7, in dynamic graph drawing there is a trade-off between local (at each time point) and global (over all time points) layout optimisation. The ViENA framework [28] and GraphDiaries [71] offer layout stability controls to allow people to optimise the layout to their needs.

Interaction techniques including filtering, clustering, grouping, and simplification [231], and network motif glyphs [232] all may help find patterns in Q2 at different levels of granularity.

8.2.2.3 Q3

For Q3, nested views e.g. [70], [73] (**Figure 71** and **Figure 89**), show temporal trends of nodes or edges embedded within the graph structure. Temporal trends can also be combined with other representations: TimeFluxes [233] (Figure 72) connect the same node in two different timeslices of a 2.5D representation, and display timelines of attribute values for individual interactively-selected nodes. Vertex small multiples [197] (Figure 73) can be selected from a matrix cube view to show connectivity patterns between individual nodes over time.

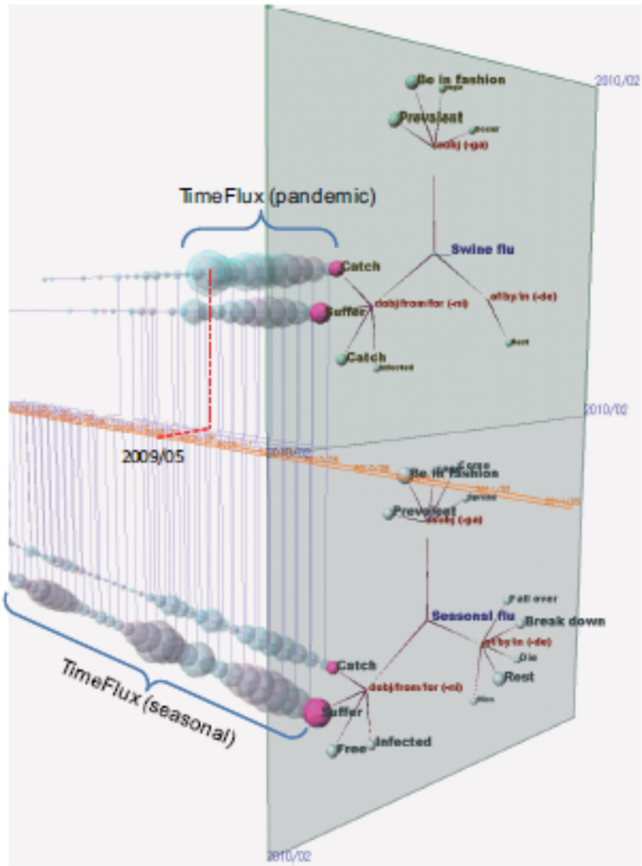


Figure 72 TimeFluxes (Itoh et al., [233]**, Figure 8) between 2.5D timeslices show temporal trends for individual nodes**

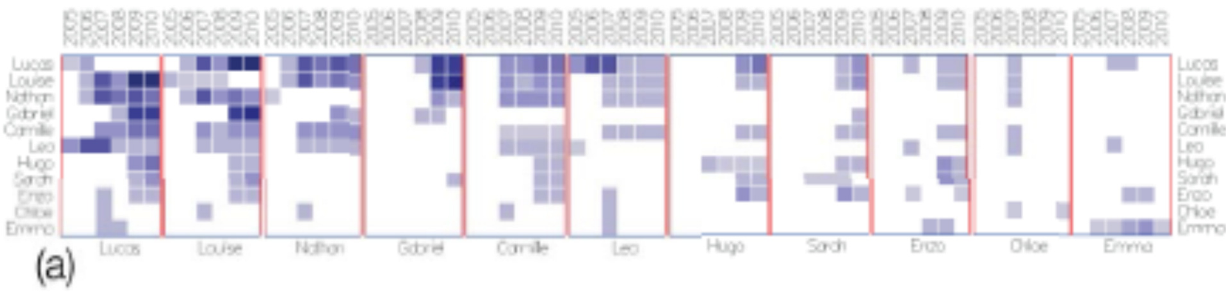


Figure 73 Bach et al.'s vertex small multiples ([197]**, Figure 8) allow comparison of temporal trends in edge weights (indicated by cell colour) between nodes**

Some systems focus specifically on showing the set of individual temporal trends. LinkWave [234] (Figure 74) visualises temporal trends in connectivity for all pairs of nodes in a graph, while NetVisia [104] displays temporal node statistics in a heatmap. Interaction techniques can be employed to filter the data to a particular time range, for example, using range sliders, as in TimeMatrix [70], or reducing timeseries data to reveal temporal patterns of interest, such as LinkWave's functionality to aggregate the temporal trends associated with groups of nodes to support the discovery of group level motifs (Figure 74). Techniques from temporal visualisation, such as ChronoLenses [235], which offers magnification and filtering, amongst other tools, and the Semantic Time Zoom techniques described in [49], [236], could also be of potential use when visualising Q3 data items.

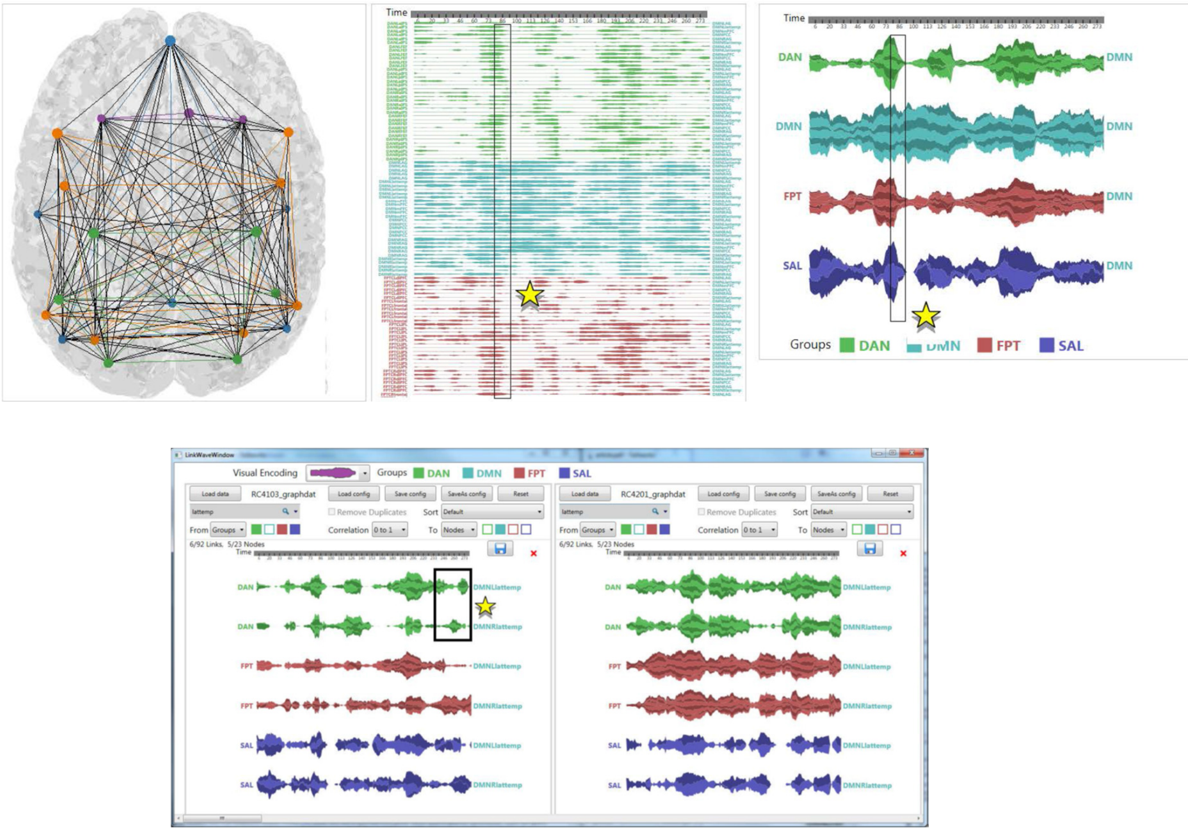


Figure 74 Riche et al.'s LinkWave ([234]**, Figure 1). Top: the connections between each pair of node at each time point is visualised using a set of streamgraphs to show the temporal evolution of the adjacency list; such a view allows comparison across temporal trends. Individual trends can be aggregated to assist in the discovery of group level connection motifs (top right). Bottom: Group level connections are compared for a diseased subject (left) and healthy subject (right) (**[234]**, Figure 4)**

8.2.2.4 Q4

When representing Q4, all of the techniques identified in the design space (Chapter 7) show graph evolution over time i.e. aspectual behaviours A4 and S4 (as discussed in Section 5.2.1), with the exception of nested views, which show the distribution of temporal trends (S1 and A1 partial behaviours) over the graph (i.e. aspectual behaviours S3 and S4).

Given the range of techniques that were identified in the design space which are able to support aspectual behaviours A4 and S4 (the changes in the distributions of attribute values over a graph, over time, and the changes in configurations of nodes over time i.e. a graph's structural evolution), careful consideration needs to be given when selecting a technique from the many available to represent such data. A number of factors may influence our choice of technique for visualising changes in a graph over time. These include:

Data considerations:

- Graph structure (general, hierarchical, compound)
- Graph characteristics (at each timeslice), such as size and density
- Overall density/sparseness of the network
- Type of change present in the data (i.e. attribute and/or structural change; node and/or edge addition/deletion)
- The amount and rate of change, or volatility of the data (e.g. many or few additions/deletions)
- Length of time series/granularity of the time dimension

Analysis considerations:

- The type of graph object under analysis e.g. paths, clusters etc.
- The aspects of the behaviour in which we are interested (e.g. our data may have node additions and deletions, but we might only be interested in nodes leaving the network)
- The granularity of pattern of interest: for example, in some cases, a topological statistic is sufficient for an analysis; in other cases, the topological structure is important. Similarly, we may require a more or less fine level of temporal granularity e.g. an aggregated view of the network for a given time

interval may be sufficient for our purposes, or we may need to see a more detailed sequence of events and show the network at each time point[33].

Section 7.4 considered the relative strengths and weaknesses of the different visual encodings, which can be of assistance when selecting a technique for a specific data set.

8.2.2.5 Search space

Depending on the task search space (discussed in Section 5.5.1), for inverse lookup tasks, only a sub section of time or graph may need to be displayed. Where the search space is time, highlighting or filtering of the graph or set of time series can be used to show only the graph object of interest. For static graphs, "Degree of Interest" techniques have been developed which show only relevant portions of a large graph (e.g. [227], [237], [238]); these could also potentially be of use when applied to temporal graphs, in the case where the search space is time. If the search space is graph, only the time period of interest need be selected and shown e.g. as mentioned in Section 8.2, Chang et al. [171] allow selection of a time interval over which to watch their sequential temporal graph visualisation unfold. Where the search space is both time and graph, showing the whole graph over all time points is necessary, and may require interaction techniques to allow people to navigate the whole dataset while searching for patterns of interest. Note that the representation used to show the data will depend on the pattern of interest. For example, if we are interested in finding nodes having interesting temporal trends in a particular attribute value, the temporal trends for all nodes will need to be displayed. Likewise, if we wish to find times at which particular structural patterns appear, we need to show the graph's structure at all time points.

8.2.2.6 Identifying time steps and graph objects

Once a pattern or value of interest is observed, the corresponding time steps and graph objects must be identifiable. Aris and Shneiderman [239] identify labelling as a challenge when representing graphs. In general there is a trade-off between being

[33] NB Bender-deMoll and McFarland [159] discuss temporal aggregation in dynamic graphs in more detail.

able to view labels, and the clutter and occlusion which showing them may cause. Showing all labels all of the time may obscure the data, especially in larger graphs. The alternative strategy is to show some labels some of the time through interaction e.g. showing labels on demand on mouse-over, or use of a Degree of Interest function to determine which labels should be displayed. Labels can be displayed in situ (next to the node/edge) or in a separate area of the screen (e.g. a side window). Fekete and Plaisant [240] offer a taxonomy of general labelling strategies in visualisation, which they divide into static (e.g. showing labels only when there is sufficient space) and dynamic techniques (e.g. tool tips or display in side window on mouse over, excentric labelling), and offer advantages and disadvantages of each. Similar labelling strategies (show all labels or only show labels on-demand) can be employed where a timeline is present, or individual timeslices can be time stamped in sequential views e.g. [172], [241] (Figure 69 and Figure 75).

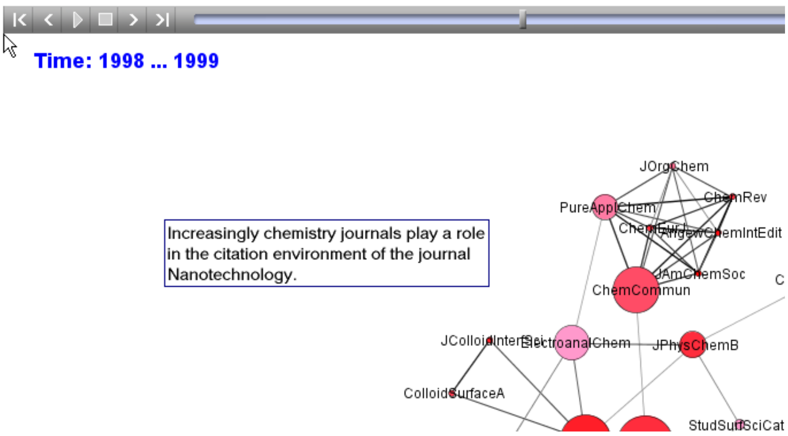


Figure 75 Example of time stamped labelling (top left). Screenshot from http://www.leydesdorff.net/journals/nanotech/

As tasks may be chained, some way of marking found graph items and/or time points for use in subsequent tasks can be supported, for example, nodes of interest are often highlighted via selection mechanisms for tracking over time e.g. [28], [71], [173] (Figure 76).

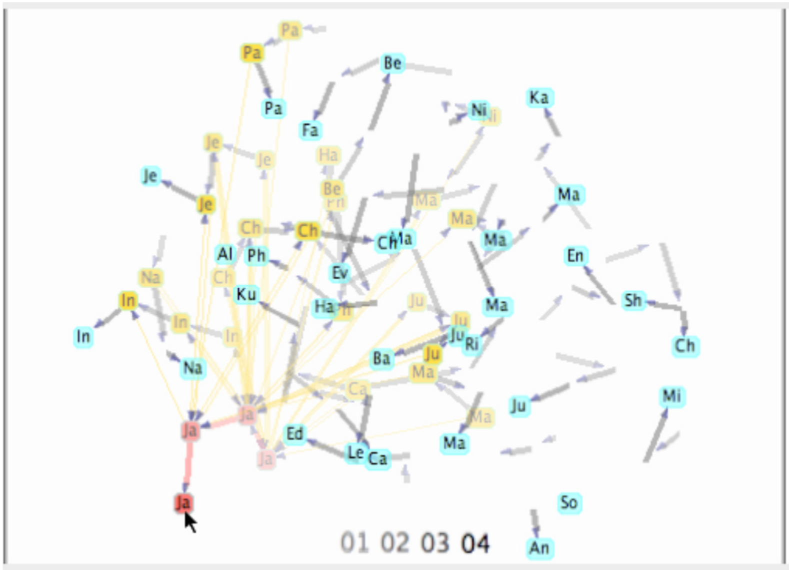


Figure 76 Superimposed view (Federico et al., [28]**, Figure 3). The selected node is highlighted in red in all four time slices.**

8.3 **Comparison**

The visual techniques appropriate to support comparison tasks depend largely on what is to be compared - graph elements or objects, time points or intervals, attribute values or patterns, or structural patterns. These are distinguished in the quadrants, and according to the direct/inverse task distinction. In many cases, at least one of the items is found via a lookup task, which means the techniques for locating time, graph and/or patterns or values (discussed in Section 8.2) must be appropriate.

Whether the same or different graph objects, times, and attributes, or a specified item, are involved in the task also needs to be considered, as some mechanism for selecting different objects for use in comparison is required. This is discussed further in Section 8.3.3.

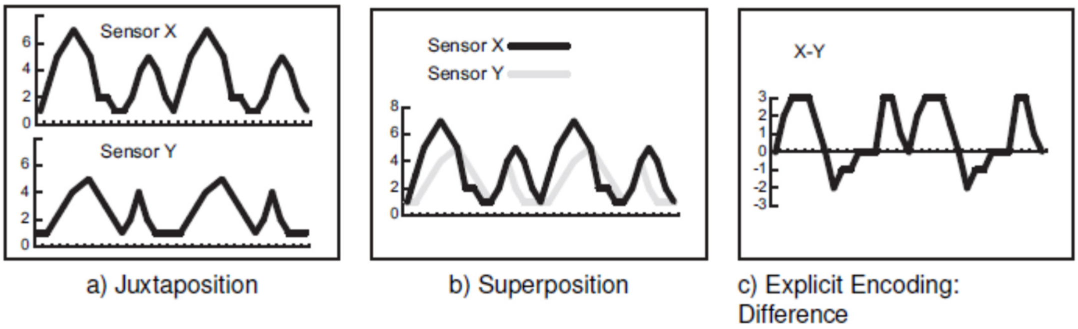


Figure 77 Gleicher et al.'s three basic possibilities for visual comparison ([63], Figure 1).

Gleicher et al.'s [102] three basic possibilities for visual comparison (Figure 77) - juxtaposition (placing representations side by side), superposition (overlaying representations in the same display space) and explicit encoding (where the relationship between the two items is calculated and explicitly represented) – can be applied in each of the quadrants. In addition, temporal graph visualisation is heavily related to graph comparison [79], and Q2 can draw on a large body of literature in this area.

8.3.1 Direct comparison

Direct comparison involves comparison of attribute values or patterns, or structural patterns.

8.3.1.1 Q1

With regard to comparing attribute values associated with individual nodes and edges, the context in which objects appear can affect our perception of them (notable examples include Adelson's illusion of colour perception[34]), with precise judgement being easier if objects are positioned next to each other and aligned. However, the position in which attributes appear in a graph representation is determined by the graph's structure. The ability to manually adjust the layout (e.g. through dragging nodes to new positions) could be useful for smaller graphs. However, in large graphs where nodes are very distantly positioned or do not appear in the same screen space (e.g. where interaction is utilised in very large graphs), this

[34] See http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

may not be possible. Functionality to select nodes or edges for use in comparison views could also be helpful when comparing attribute values in Q1.

8.3.1.2 Q2

Layout, transitioning, differencing, and matching techniques can all be used to support graph comparison.

Graph layout can facilitate comparison by minimising movement in node positions between graphs [242]. In timeslice views, local layout stability in dynamic graph algorithms focusses on minimising unnecessary change between consecutive timeslices in order to preserve the mental map of the person using the system. Where the graphs being compared are from distant parts of the timeseries, the layouts may need to be recalculated relative to one another (see Section 8.3.3.1).

In sequential views, transitioning techniques help people to follow changes occurring between timeslices, thereby supporting maintenance of the mental map. Techniques include staged transitions [243], and animation – either interpolation of node positions in node link diagrams e.g. [241], or animated changes in space-filling representations e.g. [150]. Motion ('blinking'), fading, and colour highlighting are often used to draw attention to the addition or deletion of graph elements. Bach et al. [71] note the need to be aware of interference between encoding used in transitions and the encodings representing graph data when designing transitions; their GraphDiaries system is a good example of the use of transitioning techniques.

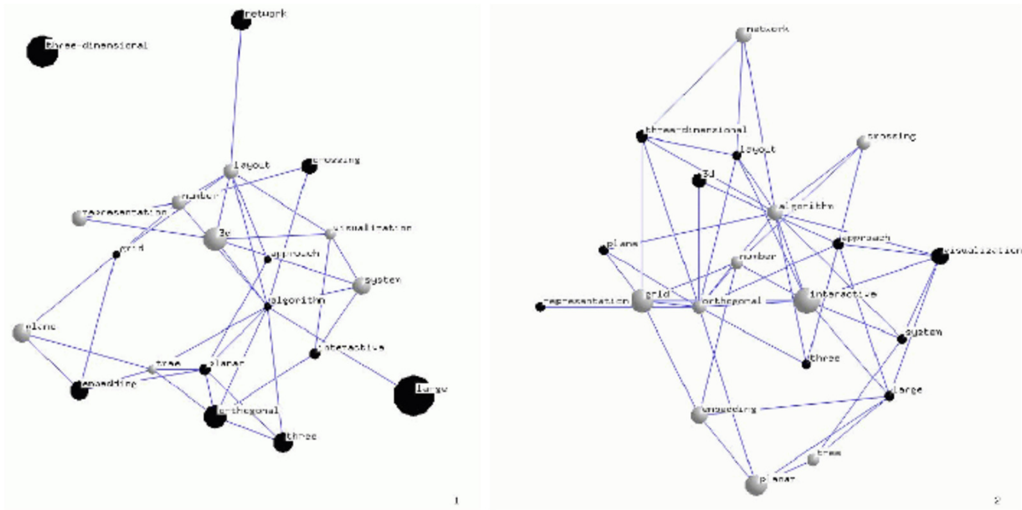


Figure 78 GraphAEL's difference graphs ([23] **Figure 5) encode magnitude of change with node size; dark nodes indicate increase, light nodes, decrease.**

Differencing techniques reflect Gleicher et al.'s [102] explicit encoding category, using visual encoding to represent the difference between two timeslices. Such techniques have been used in conjunction with both node link diagrams [23], [144], [145] and treemaps [146], and can be used to represent attribute change [23], [146] (Figure 78 and Figure 84) or structural change [144]–[146] (Figure 79). Differencing techniques may potentially be used in conjunction with sequential views, juxtaposition, and possibly 2.5D approaches to temporal encoding, although literature regarding their use in temporal graph systems (as opposed to their merits in graph comparison) is more limited; [23], [71], [145] are examples.

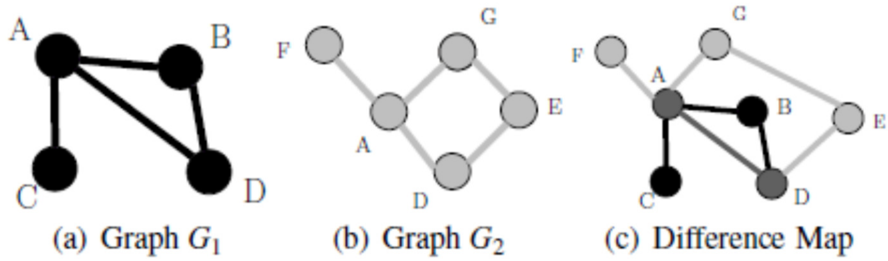


Figure 79 A difference map ([144]**, Figure 1) (c) is constructed by combining Graphs G_1 and G_2: black and light grey encode the nodes which appear only in G_1 and G_2, respectively. Nodes A and D, which are common to both graphs are shown in dark grey.**

Superposition is also used to show change between timeslices. ViENA's [192] superimposition view (Figure 76) and DARLS' [145] difference layers superimpose one graph on top of another, and indicate the different timeslices using a visual variable, such as colour or transparency. As nodes common to both timeslices appear twice, this approach shows not only the structural differences between two graphs (node/edge addition/deletion) but also change in node position.

Related to differencing techniques are techniques for highlighting *matches* between timeslices. Hascoët et al. [103] note the use of three different approaches in graph comparison: use of visual links, colour coding, and brushing and linking. Examples in the temporal graph literature include, TimeFluxes [173] (Figure 80) and node trajectories [192] (Figure 81) which link nodes in different timeslices; and co-ordinated highlighting e.g. [136], [173], which help in locating nodes and comparing their positions between timeslices.

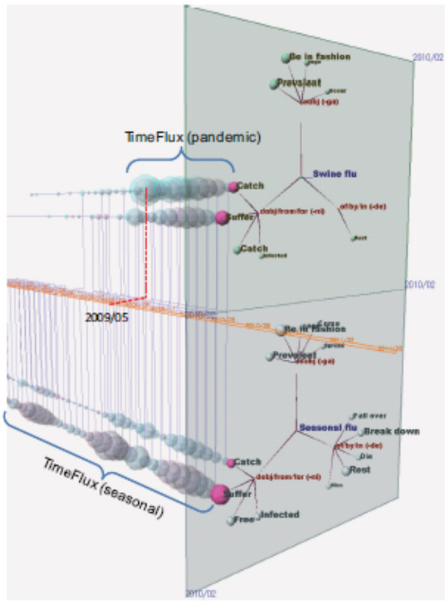


Figure 80 Comparison of TimeFluxes (Itoh et al., [233]**, Figure 8), which show temporal trends for individual nodes**

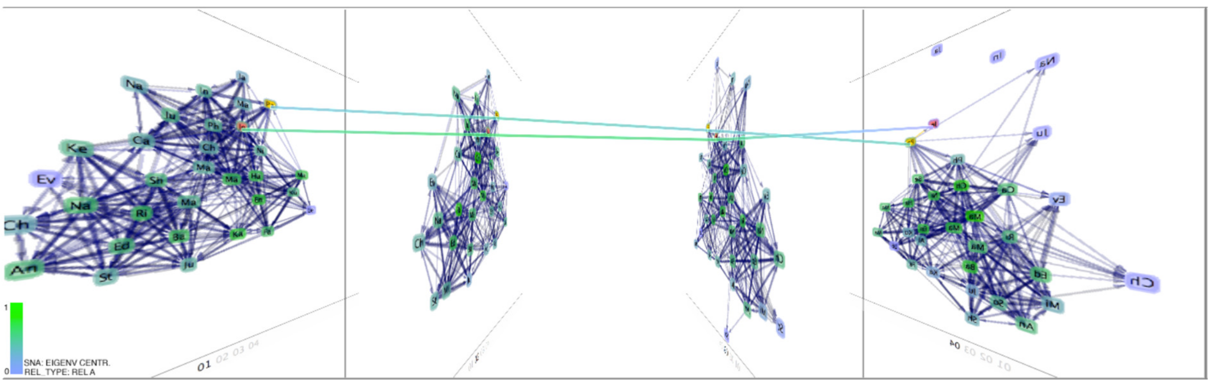


Figure 81 On-demand node trajectories in ViENA's 2.5D view ([244]**, Figure 2)**

The techniques described above relate specifically to timeslice views. Other techniques supporting comparison include alluvial diagrams [137] (**Figure 82** which show significant changes in clustering between adjacent time points, and ManyNets [245] which offers tabular views for comparison of statistics relating to multiple networks.

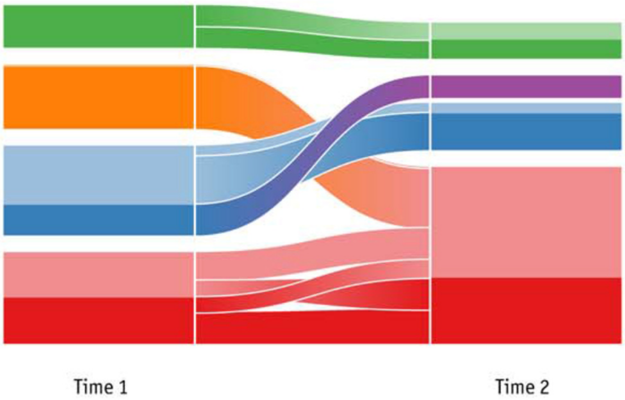


Figure 82 Alluvial diagrams (Rosvall & Bergstrom, [137]**, excerpt from Figure 2): each block represents a cluster; different colours within the same block representing significant subsets.**

In general, comparison of graph attributes is not well considered in the literature. Alper et al. [246] carried out a controlled study to evaluate techniques for weighted graph comparison using node link and matrix layouts (Figure 83), finding matrix approaches to be more effective for encoding and comparing edge weights. Some support is offered in temporal graph systems, e.g. GraphAEL [23] (Figure 78) offers a version of explicit encoding on nodes which encodes the difference in values between two timepoints as node size, while Tu and Shen [146] offer a number of techniques for showing change in attribute values between two treemaps (Figure 84).

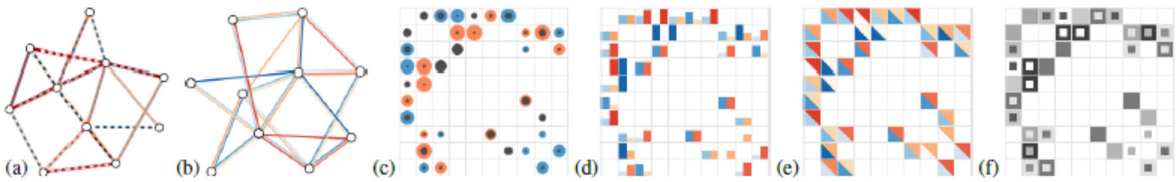


Figure 83 Techniques for weighted graph comparison considered by Alper et al.([246]**, Figure 3).**

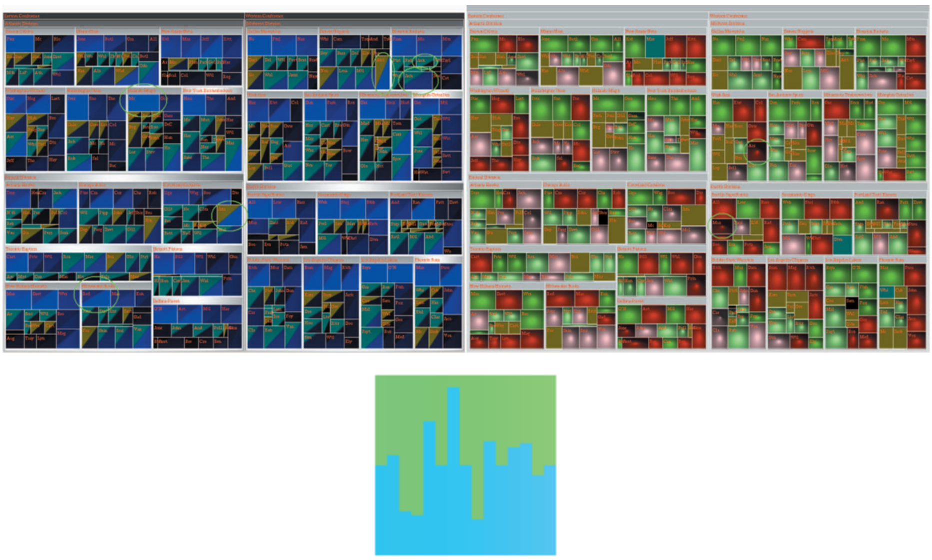


Figure 84 Tu & Shen's techniques for showing change in attribute values between two treemaps. Left: In two-corner contrast treemap ([146]**, Figure 11) the upper left corner represents time 1, lower right time 2. Middle: Ratio contrast treemap (**[146]**, Figure 14) uses explicit encoding (colour, saturation, brightness) of the ratio change between time 1 and 2. Right: They also offer a contrast treemap for multiple attributes (**[146]**, Figure 15): each vertical segment represents an attribute, the top half (green) represents time 1, bottom half (blue), time 2.**

8.3.1.3 Q3

Comparison in Q3 is little considered by the temporal graph literature. Nested views show all temporal trends for a node or edge in the same display space, which allows comparison to some extent. However, the conditions are not optimal, due to the limited display space available to show the time series, and - similar to the case of node attributes discussed in Section 8.3.1.1 - their spatial positions are determined by the graph layout. Comparison of temporal patterns is better supported where timeseries are aligned, as in LinkWave's [234] temporal patterns of dyad connectivity (Figure 74 and Figure 85) and NetVisia's [104] node attribute values. LinkWave also facilitates comparison between groups of temporal trends (Figure 85). Techniques for more flexible selection of timeseries associated with different graph objects, time periods, and attributes, for use in comparison tasks, could be considered when designing temporal graph systems (see Section 8.3.3).

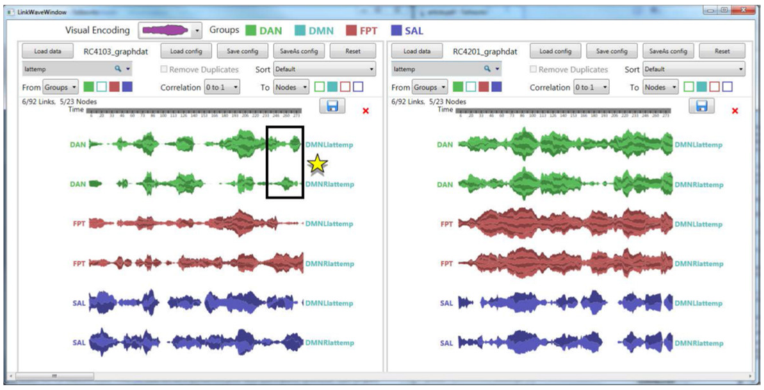


Figure 85 Facilitating comparison between groups of temporal trends in LinkWave (Riche et al., [234]**, Figure 4).**

8.3.1.4 Q4

Comparison of data items in Q4 (evolving graphs or temporal distributions over graph structures) is not well documented in the literature. Saraiya et al. [73] and Yi et al. [70] allow multiple attributes to be displayed in their timeseries glyphs, potentially supporting comparison of temporal distributions of different attributes over the graph. MatrixFlow [196] offers a juxtaposed view of the evolution of three co-occurrence matrices aligned over the same time period. Itoh et al. [173] support comparison of evolution of two *different* graphs: at each time point, a timeslice from each graph is combined in one of three ways (aggregate, pile, or split view – which reflect Gleicher's approaches) (Figure 86). These combined timeslices can then be visualised using the temporal layouts offered by their system (animation, juxtaposition, 2.5D, merged and superimposed views). An interesting direction for future research would be to adapt these techniques to explore the possibilities relating to comparison of different parts of the graph, different time periods, and different attributes (see Section 8.3.3), and also assess the effectiveness of combinations of comparison techniques and the temporal encodings (e.g. is comparing sequential (animated) views side by side an effective way to compare structural change over time?).

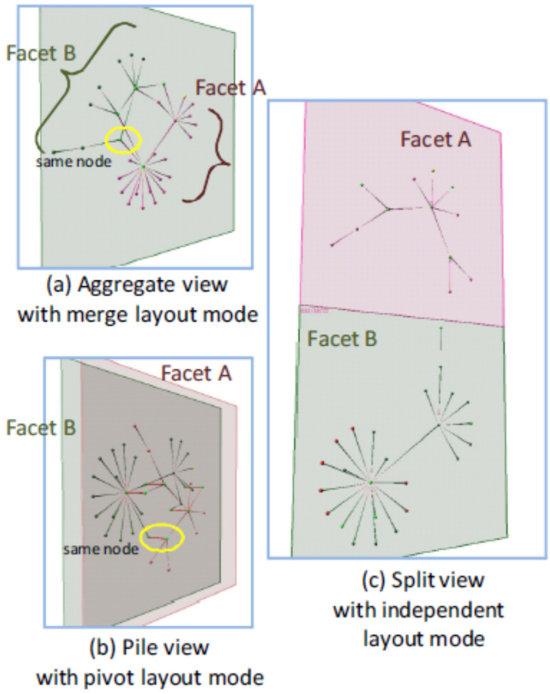


Figure 86 Itoh et al.'s ([173]**, Figure 10): three techniques for comparing selected timeslices from two different graphs**

8.3.2 Inverse comparison

Inverse comparison involves comparison of time or graph objects. In order to compare the times or nodes associated with a value or pattern of interest, these must be identifiable to the person using the tool (as discussed for lookup tasks, Section 8.2.2.6).

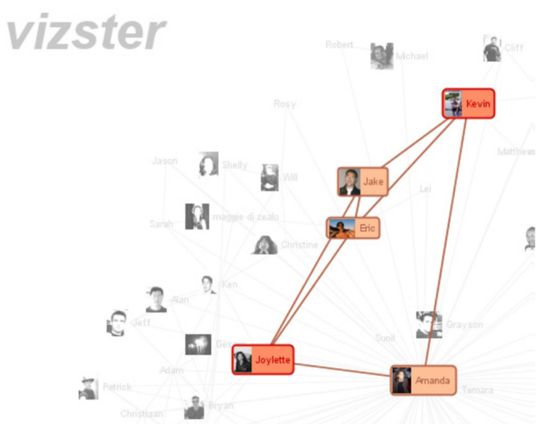


Figure 87 Vizster's Linkage View [247] **shows the intermediary nodes between two selected nodes (highlighted in red).**

Assessing the connectivity of two graph objects (elementary structural comparison) can be supported by highlighting the edge or path between two selected objects, as seen in e.g. Vizster [247] (Figure 87). PaperLens [248], a system for static graphs, allows the selection of two nodes from a drop down list, and displays the degree of separation links between them.

8.3.3 Variations in comparison task involving same or different time, graph, and attribute components

So far, the techniques for supporting comparison in different quadrants (i.e. the different types of comparison resulting from the different types of data items involved) have been discussed. Table 40 and Table 41 show the possible combinations of same and different time, graph, and attribute components, which can potentially be involved in comparison tasks. In addition, comparison may involve a specified component. This section now considers some of the ways in which these variations can be supported.

Time	Graph Component	Attribute
Same	Same	Different
Same	Different	Same or different
Different	Same	Same or different
Different	Different	Same or different

Table 40 Possible combinations of same or different time, graph components, and attributes which may participate in direct comparison

Time	Graph Component
Same	Different
Different	Same
Different	Different

Table 41 Possible combinations of time and graph components in structural comparisons

8.3.3.1 Comparison involving different times

For Q2, most temporal graph systems focus on comparison between adjacent timeslices. A few systems support comparison of non-adjacent timeslices through use of transitioning techniques [71], filtering of a small multiple display to allow juxtaposed comparisons [136], or selection of timeslices for use in comparison views [145], [173]. Once timeslices have been selected, DARLS [145] offers juxtaposed and superimposed views, and relative re-layout of graphs to facilitate comparison. Itoh et al. [173] offer juxtaposed, superimposed, and animated views, and consider methods for computing layouts in such cases. Positions of nodes in the timeslices are synchronised, and co-ordinate panning and zooming in the graph, and highlighting of nodes is employed. DGDtool [167] allows the selection and comparison of multiple timeslices and the application of different layout algorithms, allowing comparison of the *same* timeslice laid out in different ways, or comparison of two different times.

In Q3, aligning the time periods being compared may assist in making comparisons. As discussed in Section 8.3.1.4, comparison in Q4 is very limited; there are no systems which allow the selection and comparison of evolving graphs over two *different* time periods.

8.3.3.2 Comparisons in the same timeslice

While comparison of graphs at different times can be facilitated by allowing selection of timeslices for use with comparison techniques, comparison in Q2 may also involve comparison of graph objects in the same time slice. Additional support for this may be required for large graphs where the components being compared are distantly positioned in a crowded display. Techniques from static graph visualization could be employed here: DualNet [249] allows selection and comparison of two different parts of the same (static) network, in linked side-by-side views.

8.3.3.3 Comparison of two graph objects over the same time period

No techniques have specifically been developed to support comparison of different graph objects evolving over the same time period (a variation of comparison in Q4). While existing techniques which show the evolution of the graph over time may allow such comparisons to be carried out manually i.e. through visual inspection, adapting the techniques described in Section 8.3.3.2 (such as selecting the graph objects of interest) to temporally evolving graphs could prove useful here.

8.3.3.4 Comparison of different attributes

As previously mentioned, little attention has been paid to comparison of graph attributes in the literature. Even less attention has been given to supporting comparison of *different* attributes e.g. comparing the distribution of attribute A with attribute B, or comparing the evolution in distributions of attribute A and B over the graph, over a particular period of time.

Comparison of different attributes is common in temporal visualisation, where two different attributes can be charted, for example, on the same line graph or in a stacked bar chart. The nested views are therefore perhaps more readily able to incorporate such functionality. For example, TimeMatrix [70] supports comparison of the temporal behaviour of two different types of edges between the same pair of nodes, or comparison of different attributes over time for an individual node or edge, using overlays (Figure 88). Saraiya et al. [73] investigated a similar technique which combines heatmaps and line charts to show different node attributes over time (**Figure 89**); however, they found that the number of attributes displayed in their node-glyphs affected the accuracy of participant response.

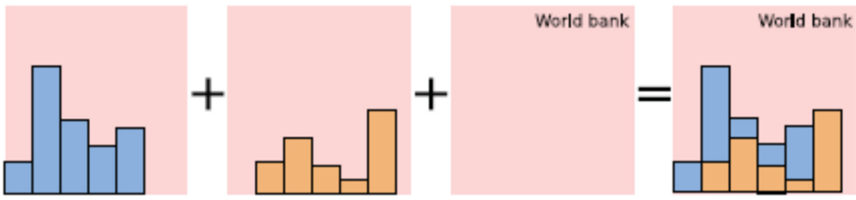


Figure 88 Illustration of TimeMatrix's "overlays" functionality which allows two different timeseries to be overlaid in a single glyph (Yi et al., [70]**, excerpt from Figure 3).**

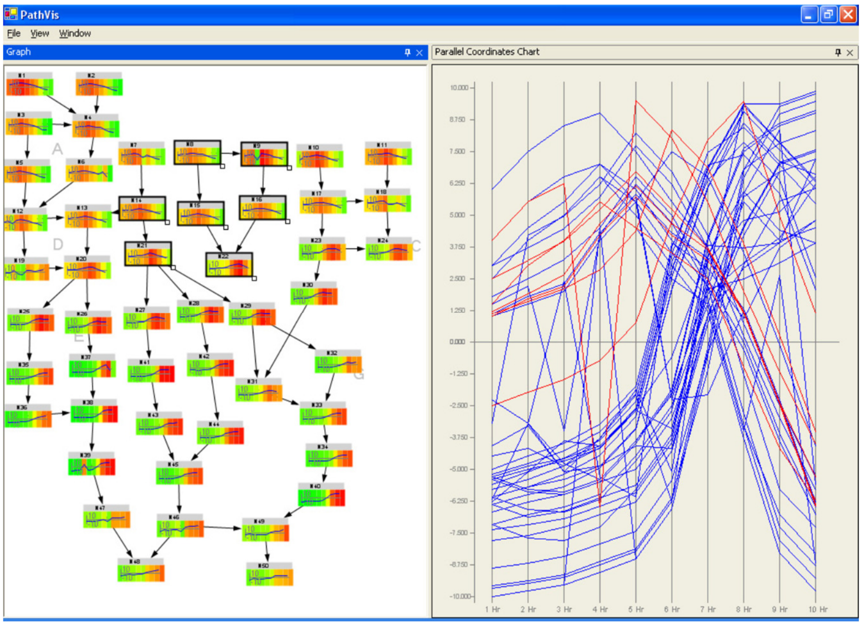


Figure 89 Heatmaps and linecharts combined in node glyphs (Saraiya et al., [73] **Figure 8)**

8.3.3.5 Comparison with a specified value or pattern

One final variation of comparison task is where a specified value or pattern (i.e. one not necessarily found in the data) is involved, e.g. a particular graph motif, temporal trend, or pattern of graph evolution. In this case a system may need some way to visually represent this for use during analysis.

8.4 Relation Seeking

Relation seeking is the opposite of comparison, in that we want to find items - graph objects, times, attribute values, patterns - related in a given way. Many of the comparison techniques also support relation seeking. Matching techniques - which find common elements between two graph representations (as discussed in Section 8.3.1.2) - can be considered relation seeking techniques in Q1 (i.e. finding the same node at two different time points). Finding nodes or edges with similar/different/opposite attribute values is generally only supported via visual inspection of the encodings used in the graph.

In Q2, Dunne and Schneiderman's [232] automatically generated network motif glyphs may help support finding similar or opposite structural patterns visually. von Landesberger et al. [250] describe a system which uses automated analysis to detect occurrences of user-specified graph motifs (either selected from a predefined list, or arbitrarily specified). We can imagine a system which might take this process one step further, and allow an analyst to interactively select a particular structure in the graph to use as the basis of an arbitrarily specified pattern upon which to detect matching patterns. Tools for relation seeking involving attribute distributions (for example functionality to find a similar or opposite distribution of attribute values to that associated with a given set of nodes) have not yet been explored.

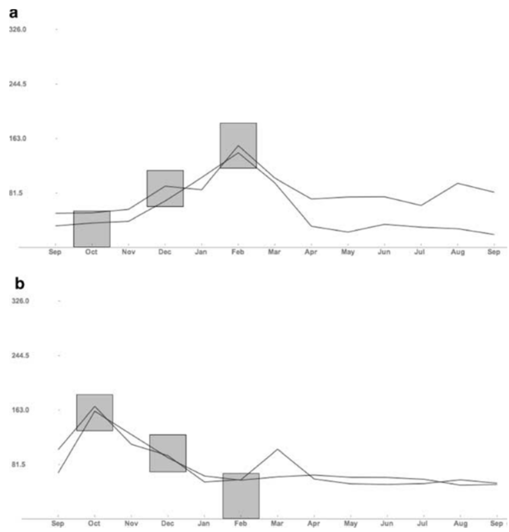


Figure 90 Finding opposite temporal trends using TimeSearcher (Hochheiser & Shneiderman, 2004, Figure 6)

TimeSearcher [114] (**Figure 90**) is a good example of a technique supporting relation seeking in Q3: specifying a slope and tolerance results in all timeseries with a similar slope being selected. A challenging opportunity for future research could be the development of similar visual analytics tools to find structural patterns in Q4 e.g. finding similar structural patterns of graph evolution or attribute distributions over time.

8.4.1 Elementary structural relation seeking

The elementary structural relation seeking task involves finding graph objects connected in a given way e.g. '*find the nodes connected to node A'.* Highlighting nodes linked to a selected node through use of e.g. colour, brightness, size, or oscillatory motion [251], is a common technique to support this task. Vizster [247] is a good example of the use of such connectivity highlighting. When a node is selected, directly connected nodes, and nodes at two degrees of separation are highlighted using a graded colour scale. Selecting an edge in Constellation [252] highlights the pair of nodes which it links. This represents support for the variation of relation seeking where no node is specified.

A further example of structural relation seeking where no node is specified is Van Ham et al.'s Phrase Net system [253] which allows the person using the system to define the relationships on which a graph is constructed. The data involved is unstructured text. The person using the system selects a relationship between words (either based on user constructed regular expressions, or by selecting a syntactic relationship from a menu) to define the edge set. A graph is then constructed in which words are nodes, and the edges between words are representative of an instance of the defined relationship. Examples given in the paper of orthographic linking (which uses text based pattern matching defined by regular expressions to construct links associated with language rules) are defining an edge (X,Y) for each occurrence of "…X's Y…" or "…X at Y…" in the data set (e.g. "King's daughter" or "dance at Netherfield"). The resulting directed graphs are then visualised using a variety of techniques in order to produce a readable graph.

8.5 Overview of the task-technique mapping

Table 42 gives an overview of the techniques which have been discussed in this chapter, according to task type and quadrant. Cells marked with a green star indicate areas where opportunities for further research were identified.

	Q1	Q2	Q3	Q4
LOOKUP	Appropriate visual encodings:			
	Determined by graph representation; attribute encodings	Graph vis; timeslice views	Temporal vis; nested views; Time Fluxes [173], Vertex Small Multiples [197], LinkWave [113], NetVisia [104]	Temporal graph vis; design space [254]
Direct (‘*find attribute values or patterns, or structural patterns associated with given graph objects at given times*’)	Graph and temporal navigation			
Inverse (‘*identify graph/time components corresponding to attribute values or patterns, or structural patterns*)		Filtering and reduction techniques to reveal patterns		
	Filtering/highlighting to reduce search space			
	Labelling strategies to identify time/graph objects			
	Marking found graph objects/times for use in later tasks			
COMPARISON	Gleicher’s approaches [102]: juxtaposition, superposition, explicit encoding			
	Display a specified data item			
Direct (‘*compare attribute values or patterns, or structural patterns’)*	Alignment, colour context Graph comparison techniques – layout, transitioning, differencing, matching; co-ordinated pan & zoom		Nested views; aligned timeseries	Examples: [255], TimeMatrix [70], MatrixFlow [196], [173]
Inverse (‘*compare (find the relationship between) graph objects or times’*)	Identifiable graph/time labels			
	Interactive highlighting of connections between selected graph objects; PaperLens [248]			
RELATION SEEKING (‘*find data items related in a given manner*’)	Matching techniques (visual links, colour coding, brushing and linking); interactively highlighting nodes linked to a selected graph object; Phrase Nets[253]; Graph motif matching [250]		TimeSearcher	

Table 42 Summary of techniques supporting tasks types in the four quadrants. Possibilities for further research mentioned in the discussion are highlighted with a star.

8.6 **Combining Techniques**

The above discussion has considered techniques for the support of individual task types. However, many individual tasks of varying types are involved in exploratory analysis. Moreover, depending on the pattern of interest, the exact same task may be best supported by different visual representations e.g. different layout algorithms draw attention to different structural features (clusters, hierarchy, etc.), while aggregating time or graph structures reveal patterns at different levels of granularity. Further, as noted in Section 8.2.2.6, tasks may be chained, with the result of one task being the starting point for the next (e.g. having found a graph object with a particular attribute value or interesting structural feature, we may then want to observe how it evolves over time). Andrienko [5] also highlight the need to synthesise the findings from our partial observations in order to form a coherent view of the overall behaviour of the data.

A variety of tools are therefore necessary to support exploratory analysis. These different tools must be integrated in such a way as to fully support an iterative analysis process, and allow the person performing the analysis to piece together their partial understandings of the data. This section therefore considers the ways in which different techniques can be combined, and the ways in which tools can support the integration of findings.

8.6.1 Multiple views

The importance of offering multiple views on the data in order to maximise insight [256]–[258], balance the strengths and weaknesses of individual views [39] and avoid misinterpretation [259], is a well-established design principle in visualisation. There are two general possibilities when offering multiple views: the person using the system can be offered a choice of ways to represent the data which they can switch between, or views can be combined in some manner in the same display space. Co-ordinated multiple views (CMV) not only combine visual representations in the same display space, but use co-ordinated interaction techniques. Often views are juxtaposed side-by-side, and interacting with one view results in some change in another view, such as highlighting a corresponding item, or zooming or filtering the views in a co-ordinated manner. Javed and Elmqvist [39] identify additional ways in which views can be combined, and introduce five design patterns for what they term “composite visualisation views” (CVV) (illustrated in **Figure 91**). Note that these patterns were utilised when determining the possibilities for the temporal encodings of the design space in Chapter 7.

Figure 91 Four of Javed and Elmqvist's composite visualisation views (CVV) design patterns (left-right): juxtaposition, superimposition, overloading, nesting (Javed & Elmqvist, 2012, Figure 1). They also include a fifth pattern, integration, which involves the use of visual links between views.

8.6.2 Multiple views in temporal graph systems

Let us now consider the ways in which multiple views have been combined in temporal graph systems.

8.6.2.1 Selecting a different representation

A number of temporal graph systems allow the person using the system to select and switch between representations. Systems offering different temporal encodings include:

- GraphAEL [23] - sequential, juxtaposed and 2.5D views;
- Cubix [197] - juxtaposed, 2.5D, and merged views;
- ViENA [28] - juxtaposed, 2.5D, and superimposed views;
- Itoh [173] - sequential, juxtaposed, 2.5D, merged and superimposed views.

As discussed in Section 8.2.2.2, systems may also offer a selection of different layout algorithms for application in node link diagrams. Interestingly, no system exists which offers switching between different graph representations (i.e. a choice of node link, matrix, space filling). However, Hadlak et al. [82] support this with their in-situ technique, which allows different temporal and graph encodings to be embedded in a base visualisation (Figure 92).

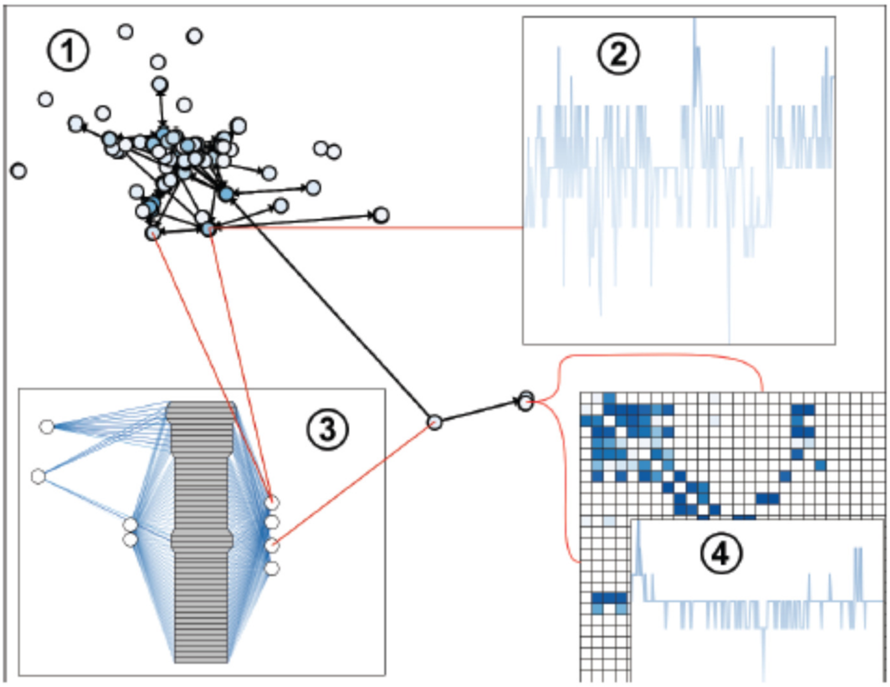


Figure 92 Hadlak et al.'s in situ strategy: "1: base visualization showing a node-link layout of the supergraph and multiple embedded visualizations. 2: in situ visualization showing a complexity plot for the underlying subgraph. 3: in situ visualization showing a 1.5D visualization of the underlying subgraph, connecting links are overlaid in red by the base visualization. 4: recursive use of in situ visualization to show a complexity plot for a subgraph selected in a matrix view." (Figure 1, [82])

At least two systems offer the person using the system a choice of views for comparing time slices: DARLS [145] offers juxtaposed and superimposed views, while Itoh et al. [173] offer juxtaposition, superimposition, and difference maps (i.e. explicit encoding).

Offering different views to support different tasks, such as switching between a "lookup view" (e.g. a view showing Q4 data items, such as a 2.5D view) and a "comparison view" (e.g. a view suitable for Q2 comparison of two timeslices), is supported by a number of systems, such as those discussed in 8.3.3.1.

Federico et al. [192] note the importance of supporting the mental map when switching between views. Their 'vertigo zoom' interaction technique does this through use of smooth transitions between the structural and temporal aspects of the data. Similarly, Cubix [197] animates transitions between views to maintain user understanding. A "Cubelet" widget acts as a visual metaphor and interactive controller, which represents the current and possible views offered by the system.

8.6.2.2 Composite views

Most of the techniques for visualising temporal graphs identified in Chapter 7 already utilise composite views: as mentioned above, the temporal encoding of the design space incorporates Javed and Elmqvist's CVV patterns [39]. In addition, some systems have purposefully incorporated more than one graph and/or temporal encoding in the same view.

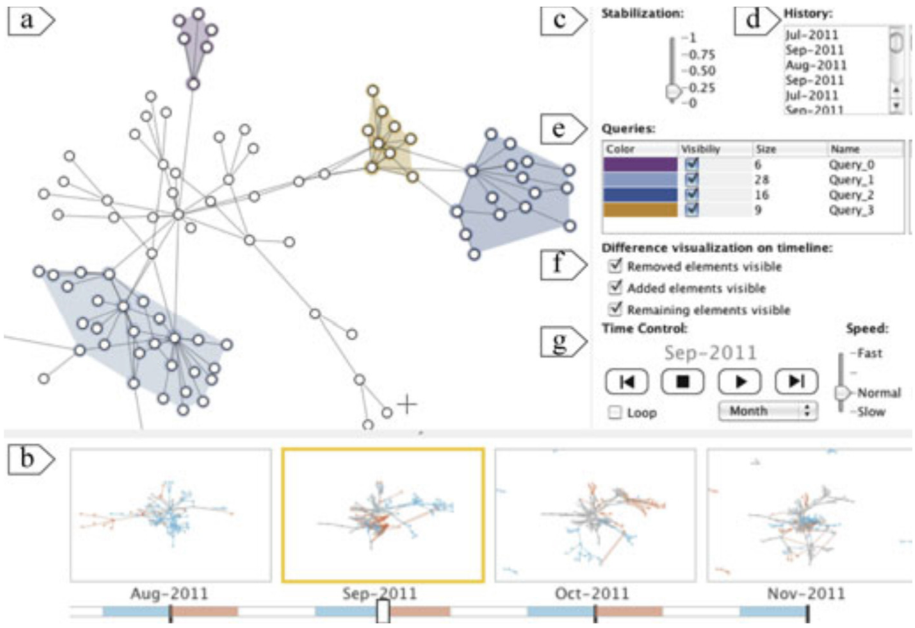


Figure 93 GraphDiaries combines sequential (a) and juxtaposed (b) temporal encodings in the same display space ([71] Figure 1)

Different temporal encodings can be displayed in the same screen space, for example, GraphDiaries [71] (Figure 93) combine sequential and juxtaposed views. As mentioned in Section 8.2.1, timeline views of statistical summary information are frequently shown together with sequential views.

Systems which allow the person using the system to interactively select views and show them in the same screen space include DiffAni [83], which incorporates small multiple, animation and difference map 'tiles' which can be selected for different parts of the timeline, and Hadlak et al.'s [82] 'in situ' technique (discussed in the previous section), which allows multiple views of both the temporal and graph structural aspects of the data to be selected and shown together in a single, tightly integrated view. Itoh et al.'s [173] 2.5D views combined with interactively selected

TimeFluxes showing temporal information relating to individual nodes, are also good example of this type of composite view.

8.7 Piecing together findings

So far we have considered the techniques which support different tasks, and ways in which these different techniques can be combined. Support for integrating partial findings is also required. The "process & provenance" category of Heer and Shneiderman's [59] interaction taxonomy is relevant here. In particular, they consider techniques for recording 'interaction histories' and annotating findings. Wybrow et al. [260] review three systems offering means to record a person's interactions with *multivariate* graphs: GraphDice [261], RelaNet, and CZSaw [262]. However, they conclude that this remains a large challenge in visual analytics generally.

8.8 Role of the task classification in task-technique mapping

This chapter has considered the techniques to support different tasks involved in exploring temporal graph data, and identified a number of areas where support for tasks is lacking. Let us now consider this finding in relation to the task classification used as the basis of task-technique mapping, specifically with regard to the findings of the evaluation outlined in Chapter 6.

While evaluating the task classification, it was found that

(1) It was more comprehensive than extant task classifications
(2) Some further work was required to establish the real world nature of some categories of tasks.

Table 43 summarises the overlap between the task categories identified as opportunities for further research in this chapter (marked with a star) and the findings of the evaluation relating to task coverage in existing frameworks in Chapter 6 (blue shaded area highlights categories identified in the task classification of this thesis which are not found in extant classifications).

	Lookup	Comparison	Relation Seeking
Q1		*	*
Q2		*	*
Q3		*	
Q4i		*	*
Q4ii		*	*

Table 43 Summary of task categories identified in this chapter as opportunities for further research into visual techniques for their support (marked with a star) and task categories appearing in the task classification developed in this thesis, but not in extant classifications (shaded in blue). Large stars indicate categories not appearing in other task classifications and identified as opportunities for research.

Firstly, we can conclude from this that there are a set of tasks (relation seeking in Q4i and comparison and relation seeking in Q4ii; marked with a larger star in Table 43) which could benefit from further research into techniques for their support which it would not have been possible to identify by performing a task-technique mapping with the set of tasks drawn from extant task classifications. This underlines the utility of the work carried out in this thesis.

However, secondly, we should recall the discussion in Section 6.2.2.4 relating to the real world nature of tasks. While evidence has been provided at a high level in favour of the real world nature of the tasks of the classification, further work is needed to establish this for all variations of task. Where a task is identified as an opportunity for further research as it is not currently well supported, before developing techniques, we should be sure to consider the evidence in support of whether it is real world in nature, either as identified in this thesis, or from further consideration of potential domains and analysis scenarios.

8.9 Summary

This chapter has considered the techniques to support different tasks involved in exploring temporal graph data. One of the main - and perhaps surprising - findings is the need for techniques not only from research specific to temporal graph visualisation, but from the static graph and temporal visualisation research areas. The distinction between techniques which support the different aspectual behaviours – A3 and S3 (evolving attribute distributions or graph structures) and A4 and S4 (temporal distributions over the graph) – is also important, as to date these (along with any other technique used in conjunction with temporal graph data) have all simply been considered together as "temporal graph visualisation techniques". However each class of techniques is able to represent only one aspect of the data.

A number of areas for future work were identified. With regard to different task types, very little work has been undertaken to support comparison and relation seeking in Q4, and this area is ripe with possibilities for future research. More generally, offering mechanism for selecting data items for inclusion in "comparison views" to support the variety of combinations of graph, time, and attribute components which may participate in comparison tasks, should be considered when developing future systems. It was noted that some of these areas would not have been identified as areas for future work had the extant task classifications been used as the basis of the task-technique mapping. However, the need to consider the evidence in support of the real world nature of tasks before embarking upon further research was also highlighted.

Finally, we are beginning to see the emergence of systems which combine multiple techniques in different ways. There is scope for further work in this area, for example, in developing mechanisms for re-using results; including techniques from all quadrants to support the different task types and developing mechanisms to switch between views; and "process and provenance" techniques to help track exploration history and integrate partial findings from the analysis process.

Chapter 9 Case Study

This chapter presents a case study in which the tools developed in this thesis are used to evaluate an existing temporal graph visualisation system. The system is evaluated in terms of the tasks which it currently supports, and a number of unsupported tasks are identified. Based on this, recommendations for the inclusion of additional tools can be made. The intention of this chapter is to demonstrate one way in which the tools in this thesis could be used in the design and evaluation of temporal graph systems. The tools could also be utilised in a similar way in the design of a new system. The other uses of the task design space and task-technique mapping which centre around identifying research opportunities, for example, identifying unexplored visual techniques, unsupported tasks, and opportunities for evaluation where multiple techniques support a single task, etc., are demonstrated in Chapter 7 and Chapter 8.

As noted in Section 2.3, when designing and evaluating systems, taxonomies are not intended to be used in isolation from the input from the people who will use those systems; they provide an additional tool to point to tasks and techniques which may otherwise be overlooked. Note that the case study in this chapter is limited to the stages directly involving the use of the task taxonomy and task-technique mapping. The next stage would be to discuss these findings with those for whom the system is intended.

The chapter is organised as follows: in order to make the task framework more manageable to work with during evaluation, a methodology for its use was developed. This is presented in Section 9.1. The features of the existing visualisation system being evaluated are discussed in Section 9.2. In Section 9.3, the system is evaluated using the proposed methodology, and a number of tasks are found to be unsupported. Based on this, the main recommendations for a redesign of the tool are given in Section 9.4.

9.1 Methodology

In order to evaluate a temporal graph visualisation system, a methodology was constructed based on the dimensions of the task framework, which closely follows the structure of the task-technique mapping outlined in Chapter 8. The methodology consists of a checklist of items to consider when evaluating an existing temporal graph system:

1. Consider which of the eight behaviours are visualised. Can the time points/periods and graph objects associated with these behaviours be determined? Consider search space e.g. is an overview of the partial behaviours offered? How are individual attribute values represented?
2. Consider the functionality for selecting a particular node or set of nodes, and time point or period, in order to facilitate direct lookup/behaviour characterisation tasks.
3. Consider which data items – individual attribute values, attribute behaviours, structural behaviours, time points/intervals, graph objects - can be compared. Consider variations according to the same or different times, graph components and attributes[35] (see **Table 20** and **Table 21** for possibilities); a checklist is given in Figure 94. Is comparison with a specified data item (such as a particular attribute value, temporal trend, structural motif etc.) supported? Can the times and graph objects associated with particular values/patterns be compared?
4. Consider support for structural comparison (finding in what way two nodes or graph objects are related/linked) and relation seeking (finding nodes or graph objects related in a specified way).
5. Consider support for relation seeking between structural patterns. Consider support for relation seeking between attribute patterns.
6. Which relational behaviours are represented? In what ways can they be compared etc.

[35] Usually only attributes sharing the same domain can be compared; consider which comparisons would make sense for the dataset.

Note that the methodology could be adapted to be of use as a starting point when designing a new temporal graph visualisation system.

Comparison Checklist

The following provides a checklist of comparison tasks which could potentially be supported by a temporal graph system. Note that these tasks can be asked of each quadrant.

Attribute based direct comparison

At a single time point or interval, can we compare:

> Two different attribute values/patterns belonging to the same graph object?
>
> The values/patterns of the same attribute for two different graph objects?
>
> The values/patterns of two different attributes of two different graph objects?

At two different time points or intervals (t_1 and t_2), can we compare:

> An attribute value/pattern of the same graph object at t_1 and t_2?
>
> An attribute value/pattern of a graph object at t_1 with a different attribute value at t_2 (where the graph object is the same in both cases)?
>
> The values/patterns of the same attribute for graph object g_1 at t_1 and graph object g_2 at t_2?
>
> The values/patterns of an attribute of graph object g_1 at t_1 with a different attribute of graph object g_2 at t_2?

Comparison involving structural patterns

Can we compare:

> Structural patterns of two different graph objects at the same time point or over the same time interval?
>
> Structural patterns of the same graph object at two different times or over two different intervals?
>
> Structural patterns of two different graph objects at two different times or over two different intervals?

Inverse comparison

> Is it possible to compare the times at which values/patterns occur?
>
> Is it possible to compare graph objects at the same times?
>
> Is it possible to compare graph objects at different times?

Comparison with specified items

> Is comparison with specified data items (such as a particular attribute value, temporal trend, structural motif etc.) supported?

Figure 94 Comparison task checklist

9.2 Existing visualisation tool

The Institute of Informatics and Digital Innovation (IIDI) at Edinburgh Napier University has developed a prototype visualisation system, which amongst other functions, allows visitors to the Institute's website to explore the author publications data for the members of staff within the Institute. The tool is available at http://www.soc.napier.ac.uk/~cs22/socksvis/explore15.php.

The tool was designed to allow visitors to gain an understanding of how the Institute works in terms of research. For example, visitors might be interested in finding out who the key researchers are, who collaborates with whom and how this has changed over time, or how the Institute has developed in terms of publications and research areas.

The data is similar to that used in the examples throughout this thesis: each author has a set of publications associated with them, and belongs to a research centre. Each publication is tagged with a set of keywords, and is assigned a publication type (journal article, book, conference paper etc.). A screenshot of the tool is given in Figure 95.

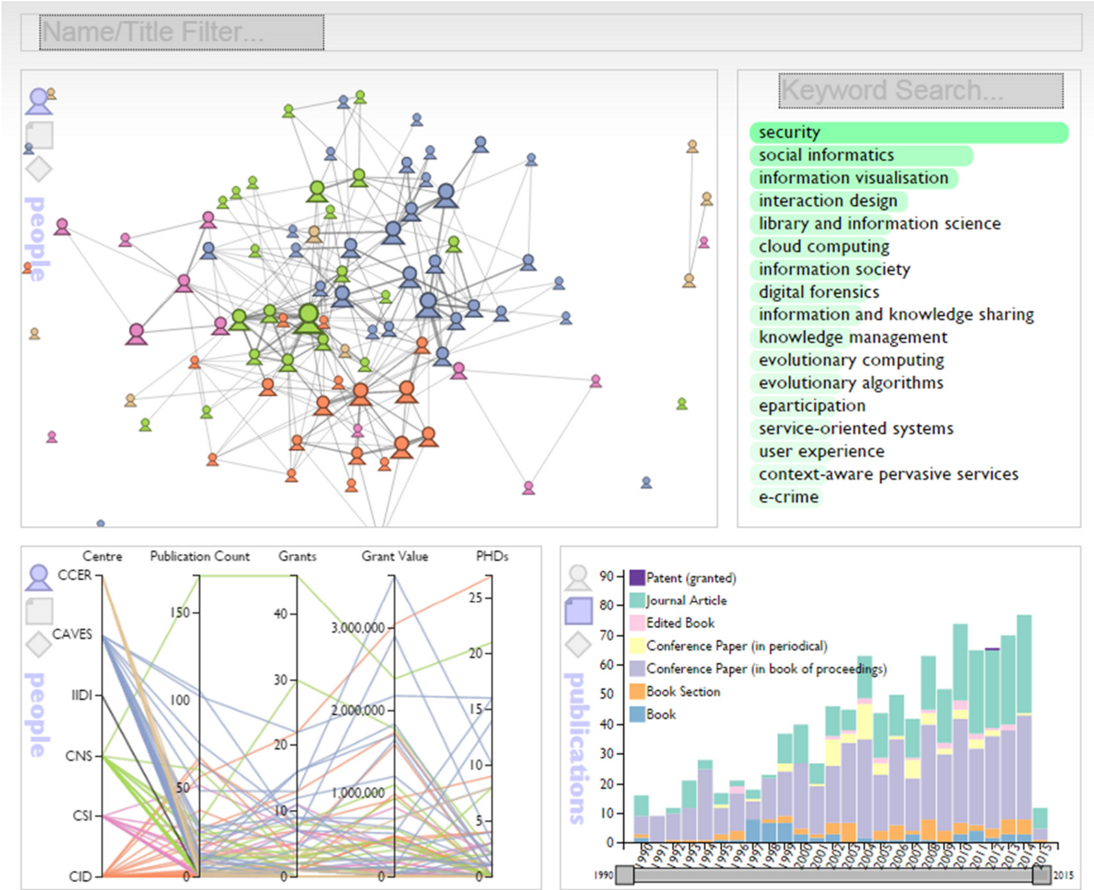


Figure 95 Screenshot of IIDI's "people" perspective, showing (clockwise from top left) a node link visualisation of the co-authorship network (which can be filtered according to the time line below the bar chart, resulting in a sequential view); bar chart of most common keywords; bar chart showing numbers of publications of each type over time; parallel co-ordinates showing various attribute values for each author. The tool can be found at http://www.soc.napier.ac.uk/~cs22/socksvis/explore15.php. Free text search facilities filter the visualisations by a person's name or a publication's keyword, and various interactive filtering mechanisms via direct selection on the visualisations are provided.

The tool consists of four main visualisations, plus various filtering mechanisms:

- **Node link co-author network** (top left, Figure 95): The node link visualisation shows the co-authorship network i.e. nodes are authors and a link is drawn where two authors have co-authored a publication. The weight of links indicates the number of co-publications, while node colour encodes research centre, and node size encodes total publication count over all time periods. The slider bar (bottom right in Figure 75) filters the graph to show only nodes and edges appearing in the selected time period. Dragging the slider results in a sequential view of the network over time[36].

[36] Note that attribute values are shown for the entire time period, and do not change over time.

- **Keyword bar chart** (top right, Figure 95): An ordered bar chart is used to indicate the most frequent keywords occurring in the data.
- **Parallel co-ordinates** (bottom left, Figure 95): Each line in the parallel co-ordinates display represents an author (coloured according to research centre); the position at which the line cuts each axis indicates the value for that attribute. Distributions of (numeric) attribute values can be seen on the individual axes. Relationships between pairs of attribute values can be more clearly seen by re-ordering the axes (e.g. there is a general relationship between grant value and number of grants: authors with more grants tend to also have higher value grants). Clusters of authors with similar values across several attributes can be seen by observing bundles of lines following similar paths (in this dataset, there are no markedly distinct bundles).
- **Publication count bar chart** (bottom right, Figure 95): the stacked bar chart shows the amount of publications over time by research centre.

9.3 **Evaluation**

The methodology outlined in Section 9.1 is used to evaluate task coverage in the system.

1. Consider the behaviours supported

At present, the node link visualisation allows us to see:

- (S2) the configuration of nodes based on the linking relations between them, at a single time
- (S4) the configurations of nodes (i.e. S2), over time e.g. the evolution of the structure of the co-authorship network over time. However, this view could be improved. Use of sequential views arguably makes gaining an overview of the changes in structural patterns somewhat difficult, as discussed in Section 8.2.2.4. Meanwhile, the lack of transitioning techniques between time points, and use of a less-than-stable force directed layout make understanding the changes between time points challenging.

- A variation of A4 (the behaviour of the distribution of the attribute values over the graph over time). The visualisation currently shows the attribute distributions over the cumulative network for the whole time period. This is because the size (publication count) and colour (research centre) do not change over time. We therefore do not gain any insight into attribute distributions over the graph at an individual point in time (A2) or how the distributions changed over time (A4).
- To a very limited extent, (S1) behaviours (the behaviour of linking relations between two graph elements over time) can be seen by focussing closely on a pair of authors and moving the time slider, however, this does not offer an optimal view of this type of behaviour.

The three other visualisations (parallel co-ordinates and bar charts) show attribute behaviours in isolation from the graph structure e.g. frequency distributions of attribute values in the data set. Each bar in the publications bar chart shows information relating to total numbers of each type of publication for all authors, at a single time point (A2), and the totals over time are shown by the set of bars (A4). The keyword bar chart shows the frequency of keywords associated with all authors in the data set, over all times (A4), or filtered for particular times (A2). The parallel co-ordinate display shows the set of aggregated total values for the whole time period, for each author (A4). When a time period is selected, the display is filtered to show only the totals for authors who appear in the network during that time (i.e. particular values for the selected time period are not shown). In addition, the parallel co-ordinate view allows us to consider relational behaviours e.g. correlations between attribute values.

Selecting an author in the node link view filters all views to show data relating to the selected author (Figure 96). On selection, the bar chart shows the behaviour of an attribute value (in this case, publication counts by type) for an individual author. This is an (A1) behaviour. In addition to this behaviour, the individual attribute values (aggregated for the whole time period) associated with the author are also shown in the parallel co-ordinates view.

Figure 96 Selecting an author in the node link display filters all displays to show only data relating to the selected author

To summarise, the current tool is able to show behaviours A1 and S2; to a more a limited extent, S4; a variation of A4 (involving graph structure), and A2 and A4 behaviours in isolation from graph structure; and, to a very limited extent, S1.

The behaviours which are not currently shown (or only shown in a limited way) include:

- (S1) the behaviour of linking relations between two nodes over time or the set of these behaviours (S3), possibly distributed over the graph
- (A1) the temporal behaviour of the research centre to which an author belongs, and the temporal behaviours of the attributes shown in the parallel co-ordinates view.

- (A2) the behaviour of an attribute over a set of nodes at a single time e.g. the distribution of publication counts or research centre affiliation over the network at an individual timepoint
- (A3) the distribution of temporal trends in attribute values over the graph
- (A4) the behaviour of the distributions of the attribute values over the graph over time e.g. the change in distribution of publication counts over the graph, over time
- (S4) the evolution of the structure of the network over time.

2. *Consider functionality for selection to facilitate direct lookup*

   As discussed in Section 8.2.1, for direct lookup and behaviour characterisation tasks, we must first locate the time and graph object of interest, in order to find the corresponding values and patterns. The tool offers time slider interaction in order to locate a time (or period) of interest, while mousing over the nodes in the node link diagram offers additional information to help identify authors. A free text search function is also available to find particular authors of interest.

3. *Consider which data items can be compared, and variations.*

Using the comparison checklist outlined in Figure 94, Table 44 considers which comparisons are supported in the existing system. Note that the only sensible comparison of different attributes for this data set relates to comparison of the numbers of different types of publications e.g. a comparison of journal article count with conference paper counts.

	Q1	Q2	Q3	Q4 graph over time	Q4 time over graph
Attribute based direct comparison At a *single time point or interval*, can we compare:					
• Two different attribute values/patterns belonging to the same graph object?	Yes. Comparison of publication counts of an individual author for different types of publications is possible using the publications bar chart	No. The distributions of publication counts for different types of publication are not shown on the node-link diagram	With difficulty. Stacked bars are not the best visualisation for comparing individual temporal trends over time.	No (as Q2 behaviours are not shown)	No
• The values/patterns of the same attribute for two different graph objects?	With difficulty. We cannot select two individuals and their associated attribute values in order to make comparisons; selecting a node only highlights the attribute information for that node on the parallel co-ordinate vis/publications bar chart, therefore comparison needs to be done in memory. Some comparison of node encodings within the node link vis (publication count=size, colour=research centre) can be made; although the lack of alignment	To some extent we can visually compare the distributions of the two attribute values encoded in the node-link diagram, although no additional assistance (e.g. selecting only the two subgraphs of interest) is given.	With difficulty. Comparison of two individuals' publication counts over time needs to be done in memory, as only the timeline for a single selected author is shown.	With difficulty. We could compare e.g. the change in distribution of publication counts over two different subgraphs over time using the timeslider, although the cognitive overhead involved in remembering the evolution of *two* subgraphs is predicted to be very high.	No

	when comparing area can impede this.				
• The values/patterns of two different attributes of two different graph objects?	With difficulty; as above such a comparison (e.g. comparing the journal article count of author A with the conference paper count of author B) would need to be done in memory.	No. The only sensible comparison of different attributes involves publication counts for different types of publication, and these distributions are not shown on the node-link diagram.	With difficulty; as above, such a comparison would need to be done in memory.	No (as Q2 behaviours are not shown)	No
At two different time points or intervals (t_1 and t_2), can we compare:					
• An attribute value/pattern of the same graph object at t_1 and t_2?	Yes. The publications bar chart allows easy comparison of an author's total publication count at two different time points. However, the attribute values in the node link diagram are aggregate totals for the whole time period therefore they do not change over time.	No. As the attribute values in the node link diagram are fixed, we cannot observe a change in distributions between time points.	Yes. We can compare the trend in an author's publication count over two different time intervals using the publications bar chart.	No	No
• An attribute value/pattern of a graph object at t_1 with a different attribute value at t_2 (where the graph object is the same in both cases)?	Yes, but with more difficulty than above. Comparison of the different types of publication counts between two different years is more difficult due to the bars not being	No. The distributions of publication counts for different types of publication are not shown on the node-link diagram	Yes, but with more difficulty than above due to alignment issues (as for Q1)	No (as Q2 behaviours are not shown)	No

	aligned on the horizontal axis.				
• The values/patterns of the same attribute for graph object g_1 at t_1 and graph object g_2 at t_2?	With difficulty. Comparison of publication counts for two different authors must be performed in memory.	No. The distributions of publication counts/research centre over the graph are aggregate totals, and do not change over time.	With difficulty. Comparison of trends in publication counts for two different authors needs to be performed in memory.	No (as Q2 behaviours are not shown)	No
• The values/patterns of an attribute of graph object g_1 at t_1 with a different attribute of graph object g_2 at t_2?	With difficulty. As above, comparison of different authors' publication counts must be done in memory.	No. The only sensible comparison of different attributes involves publication counts for different types of publication, and these distributions are not shown on the node-link diagram.	With difficulty, as above (the only difference in this task as that the comparison would involve different types of publications)	No (as Q2 behaviours are not shown)	No
Comparison involving structural patterns Can we compare:					
• Structural patterns of two different graph objects at the same time point or over the same time interval?	-	Yes, we can compare the connectivity e.g. of two clusters in the graph. However no additional assistance (e.g. selecting only the two subgraphs of interest) is given.	With difficulty. To compare the co-authoring relations over time between two sets of authors would require us to use the timeslider to step through the time and observe and remember the connectivity of each pair of authors in the node link diagram over the time series. The	With difficulty. As for Q3, we would need to remember the changes for each graph object, construct the temporal pattern in memory, and then compare them. Such a task is likely to be highly cognitively demanding.	No

			cognitive overhead of such a task is likely to be high.		
• Structural patterns of the same graph object at two different times or over two different intervals?	-	With difficulty. In a sequential view, comparing the graph at t1 with the graph at t2 is performed in memory. The lack of transitioning techniques and use of a less-than-stable force directed layout will make this task difficult.	With difficulty, as above, but we would first have to memorise the connectivity pattern over the first time interval, then memorise the pattern over the second, and then compare them.	With difficulty, as left/above.	No
• Structural patterns of two different graph objects at two different times or over two different intervals?	-	As above (although even if the tool used transitioning techniques etc. they would not help in this task).	With difficulty, as above.	With difficulty, as above.	No
Inverse comparison					
• Is it possible to compare the times at which values/patterns occur?	Yes, when using the bar chart, this is clear; when making comparisons involving the network, a time stamp in the network area could help				
• Is it possible to compare graph objects at the same times?	Yes				
• Is it possible to compare graph objects at different times?	In memory				
Comparison with specified items					
• Is comparison with specified data items (such as a particular attribute value, temporal trend, structural motif etc.) supported?	No capacity for constructing and showing a specified attribute value/trend/structure is included in the system				

Comparison of relational behaviours	Not applicable to this data set.
• Is it possible to compare relational behaviours?	

Table 44 different types of comparison supported by the current IIDI system

4. Consider support for structural comparison

Some support for structural relation seeking is supported in that selecting a node highlights the nodes to which it is connected (Figure 97). No functionality exists to specify the type of relation (e.g. a filter on edge weights).

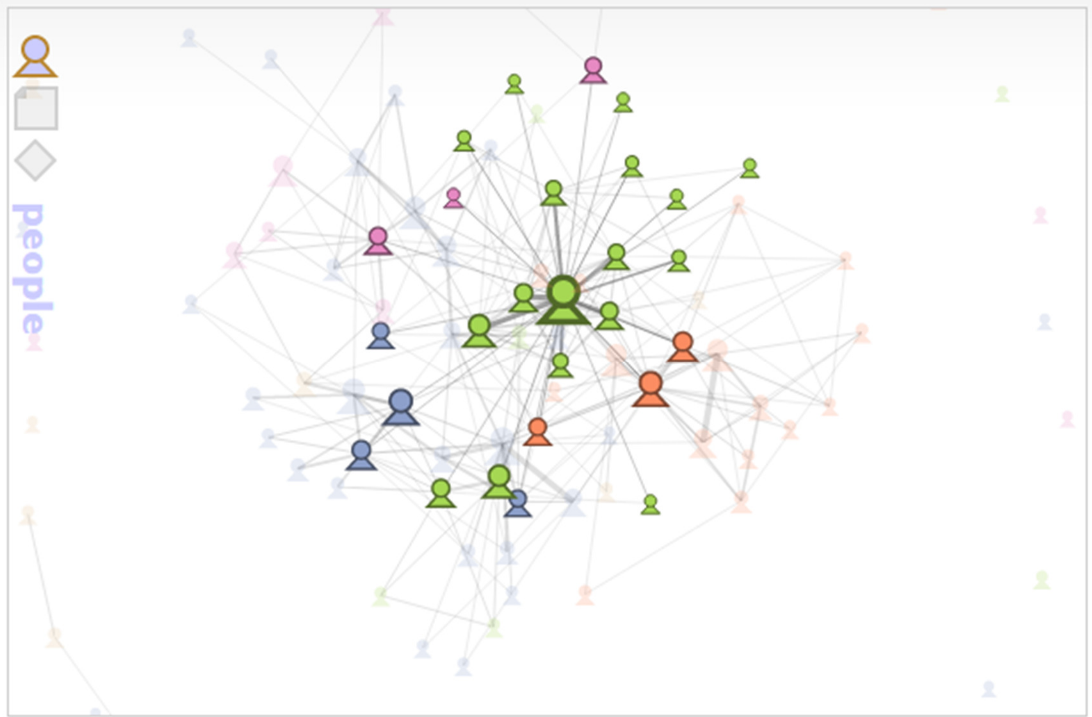


Figure 97 Selecting a node highlights the nodes to which it is connected

No specific support is offered for structural comparison (determining the relation between two nodes or sets of nodes). Such tasks rely simply on visual inspection of the node link diagram, making the identification of nodes which are not directly connected (i.e. whether a path exists between two nodes) rather difficult. Interactive zooming and the ability to reposition nodes helps in some respects with such tasks, but more support could be offered, such as highlighting the connections between two selected nodes or groups

5. Consider support for relation seeking.

Other than ‘manual’ visual inspection, support for relation seeking between structural patterns/attribute distributions (such as finding similar patterns) is not facilitated in the current system.

6. Consider support for relational behaviours

As noted earlier, the parallel co-ordinate view allows us to consider relational behaviours e.g. correlations between attribute values. The effect of graph structure on attribute value and vice versa is difficult to establish as the attribute values in the node-link diagram do not change over time. Determining the impact of particular structural patterns on patterns at subsequent times is perhaps limited by the use of sequential views which require comparison in memory and the stability issues in the layout which make comparisons between time points rather difficult.

9.4 **Main recommendations**

As the Andrienkos point out, when performing data analysis, not all aspects of the data are necessarily relevant:

"…*data analysis does not always pursue such ambitious goals as obtaining a full understanding of the overall behaviour of a phenomenon. In many particular cases, only certain aspects of the overall behaviour are relevant to the problem to be solved or only certain aspects excite the interest of the analyst."*

(Andrienko and Andrienko, [5], p106)

A general understanding of the goals and intentions of the people carrying out the analysis is therefore needed, and this requires input from those people. As noted in the introduction to this chapter, the framework outlined in this thesis is not intended for use in isolation from the input of those by whom the system will be used, but as an additional tool to point to tasks and techniques which may otherwise be overlooked. Having identified the limitations of the system, it is possible to ask the people who will use the tool whether the unsupported tasks would indeed be of interest in their analyses.

The major general recommendation for improving the system would to be offer functionality to support the eight behaviours. In particular, visualisation of behaviours S4 and A4 (changes in the graph's structure and attribute distributions over the graph, over time) could be improved by offering an alternative temporal encoding, such as juxtaposition (small multiples), and/or use of a more stable layout and inclusion of transitioning techniques in the sequential view. As the network is relatively dense, a matrix view could also be considered here.

Encoding the publication counts for each author at each time point as node size (rather than the current aggregate value) would also allow us to understand the distribution of attribute values at a single time (A2), as well as allowing us to view how these distributions have changed over time (A4). Functionality to select and view other node attributes encoded in the node-link diagram could also be useful.

The addition of a view showing the co-authoring relations between pairs of authors over time i.e. the adjacency list (such as that offered in the LinkWave system [234]) would support understanding of the S1 and S3 behaviours. Whether understanding the distribution of temporal trends over the graph is meaningful to the people using the tool in this scenario would need to be established. If it is of interest, nested views would be one way in which A3 and S3 behaviours could be supported.

Visualising the temporal behaviours of attribute values such as the research centre to which an author belongs, and/or the other attributes shown in the parallel co-ordinates view could be added to the system. Furthermore, visualising the set of temporal trends for all authors, aligned in the same display space would help us to find authors and time periods with particular patterns of interest (inverse lookup), compare patterns, perform relation seeking tasks such as finding similar - or markedly different – patterns, and gain a general understanding of the overall temporal trends in attribute values (A3).

It should be established which of the comparison tasks are most relevant to the analyses of those using the system, and implement ways to select and compare the components involved. Additional functionality to support structural comparison and structural relation seeking tasks could also be considered.

9.5 **Summary**

This chapter has presented a case study which demonstrates the use of the tools outlined in this thesis in the design and evaluation processes. In order to make using the task framework in the evaluation process more manageable, a methodology was developed. Using this methodology, a number of limitations in task coverage in an existing temporal graph visualisation system could be identified. Using the task-technique mapping, a number of recommendations to improve this system could be made. These findings serve to evidence the usefulness of the tools developed in this thesis.

Chapter 10 Conclusion

This thesis has considered the valuable role that task and technique taxonomies play in both visualisation design and evaluation, and in guiding future research in the field. Understanding the potential tasks involved in visual exploration of temporal graph data, and the possible visualisation techniques to support these tasks, have to date been only partially addressed in the literature. This work has explored "the space of the possible" for both tasks and visual techniques, through a series of taxonomies, design spaces, and mappings between these structures and existing techniques in the literature.

This chapter reflects on the original research questions posed and the extent to which these have been answered, the contributions of this work, and future directions.

10.1 Research Questions

This work has addressed the following research questions:

1. What are the possible exploratory analysis tasks that temporal graph visualisation might need to support?

2. Which visual techniques, tools, and approaches, have been developed to support exploration of temporal graph data? Are there any unexplored opportunities for visual techniques?

3. Which visual techniques support which tasks?

4. For the tasks identified in (1), are there suitable visual techniques or are new/better visual techniques required?

Let us consider in turn how these questions have been addressed.

1. *What are the possible exploratory analysis tasks that temporal graph visualisation might need to support?*

   In order to address question 1, an existing formal task framework for Exploratory Data Analysis [5] was extended to handle graph data, and a task design space was constructed to extricate the possible tasks involved in exploratory analysis of temporal graph data. This work was presented in Chapter 4 and Chapter 5. Drawing on the evaluation practices identified in the literature, the task framework was evaluated firstly in relation to extant temporal graph task classifications with respect to the properties of comprehensiveness and descriptive powers, and secondly in an empirical study primarily designed to assess its utility in the design process. This work was presented in Chapter 6, while the usefulness of the task framework in the evaluation process was explored further in the case study of Chapter 9. While further work remains to determine the extent to which the tasks of the framework are real-world in nature and its usability by visualisation researchers (see Sections 10.3.1 and 10.3.2), clear evidence in favour of its comprehensiveness and descriptive abilities were shown both in comparison to extant frameworks and in the empirical study. Its utility in the design and evaluation processes was also clearly demonstrated.

2. *Which visual techniques, tools, and approaches, have been developed to support exploration of temporal graph data? Are there any unexplored opportunities for visual techniques?*

   To answer question 2, a design space of visualisation techniques was constructed from the temporal and graph encodings identified in the literature, revealing all possible combinations. It was possible to map all of the existing techniques to the design space, indicating that the categorisations used are appropriate. The mapping revealed that the majority of existing techniques utilised node-link graph encodings and sequential temporal encodings. It also demonstrated that there is room for further research into the different possible combinations of time and graph encodings, which were less well explored. Addressing this question was the subject of Chapter 7.

3. *Which visual techniques support which tasks?*

To address question 3, the visual techniques able to support the different types of tasks were considered. This question was addressed in Chapter 8. One quite surprising, but important, finding was the need for a much wider range of visual techniques than those from temporal graph visualisation. Techniques from static graph visualisation, temporal visualisation, and visual comparison were therefore also considered. The review highlighted the need for multiple views of the data, the role of interaction in combining techniques and constructing comparison views, and the need for tools to record analysis histories and support synthesis of partial findings

4. *For the tasks identified in (1), are there suitable visual techniques or are new/better visual techniques required?*

The task-technique mapping presented in Chapter 8 also revealed a number of less well supported tasks where further research is required, answering question 4. These tasks include comparison and relation seeking in Q4, and more generally, the need to for better support for making direct comparisons for all aspects of temporal graph data. While it has been shown that some of the tasks requiring better technique support are real world in nature, in other cases it will be important to establish this before developing new visual techniques.

10.2 **Contributions**

As outlined in Section 2.3, classifications play an important role in visualisation research, facilitating communication amongst researchers, helping us make sense of what already exists in our research area and revealing opportunities for future work, and assisting in the design and evaluation processes.

Three tools have been presented in this work which are intended to be of use to visualisation researchers:

1. A classification of the potential tasks involved in exploratory analysis of temporal graph data.
2. A design space for temporal graph visualisation techniques, and mapping of extant visualisation tools to this design space.

3. A mapping between the identified types of analysis task and the visualisation tools and techniques able to support them.

Let us reflect on the development of these tools in relation to the contributions of this work that were outlined in Section 1.3.

(1) A characterisation of temporal graph data and tasks

In order to illuminate the potential tasks involved in exploratory analysis of temporal graph data, Andrienko's general task framework [5] was extended for use with graph data, and used as the basis for a taxonomy of temporal graph tasks. The dimensions of the extended taxonomy were used to construct a design space of temporal graph tasks, using a set of matrix structures to systematically capture the possible task variants.

The taxonomy and design space seek to bring structure and clarity to the range of tasks associated with temporal graphs. The main advantage of the task framework presented in this work is its comprehensiveness. While three temporal and one static graph task classification existed prior to the publication of this work, as discussed in Section 2.4.3 and demonstrated in Section 6.2.1, all of these classifications have shortcomings in terms of task coverage; none of the extant frameworks are able to capture all of the tasks of the others, and they also fall short in capturing the additional categories identified in this work. As demonstrated in the evaluation presented in Section 6.2.1, the task framework proposed in this work is able to capture all of the tasks of the extant frameworks, while also covering a number of (real-world) tasks which none of the extant frameworks had considered.

The other advantages associated with taking a formal approach to constructing the classification were outlined in Section 3.2, including:

- Tasks are specified at a consistent level of perspective, abstraction, and composition, avoiding the difficulties of abstracting tasks from concrete scenarios.
- The resultant classification is domain independent and can be of use across any discipline calling for graph visualisation.

- It not only considers tasks for temporal graphs, but provides tasks for static graphs, multivariate graphs, and graph comparison.
- The use of formal notation to describe tasks avoids ambiguity and allows highly nuanced distinctions between tasks to be made. Coupling this with verbal descriptions and concrete examples makes the tasks descriptions accessible to those unfamiliar with the formal notation.
- The formal approach allows us to explore the 'space of the possible', potentially revealing hidden tasks and corner cases, which may otherwise have been neglected from consideration had empirical techniques been employed exclusively.
- The use of task matrices in presenting the task design space allows us to see not only the nuanced distinctions between tasks but also meaningful high level categories, allowing a 'slice and dice' approach to be taken to task categorisation. This is useful, as the multiple dimensions mean that all of the tasks will fall into more than one category.

The task classification developed in this thesis is intended to be of use in assisting both designers and evaluators of temporal graph visualisation systems. The use of the task classification at the task understanding stage of the design process was evaluated in Section 6.2.2. Eliciting tasks during a requirements analysis process is a well-known problem in HCI and psychology, as people find it difficult to accurately introspect about their needs and articulate them [10], [12]. As discussed in Section 1.1, designing for Exploratory Data Analysis (EDA) compounds this problem: when carrying out EDA, the person performing the analysis may be unfamiliar with the data, and at outset, may have no specific goal in mind other than to explore and build an understanding of their data. When designing visual solutions, system designers must somehow anticipate the potential tasks in order to make an informed decision regarding which tools to include, and to ensure that a sufficiently wide range of tasks are supported. One use for task classifications is helping designers to explore the potential range of tasks that the people they are designing for might wish to carry out, but are not necessarily easily able to articulate. The study outlined in Section 6.2.2. demonstrated the usefulness of the task classification presented in this work in such a design scenario. Using the framework it was possible to discover tasks which were of interest to participants which they had not previously articulated. Further, its usefulness in abstracting and organising concrete domain tasks as the basis for selecting visual encodings was also demonstrated.

The use of the task classification in a common evaluation scenario of assessing a visualisation system in terms of its capabilities and limitations was demonstrated in the case study of Chapter 9. The task classification was used to derive a methodology to assess which tasks are currently supported by an existing system, and reveal unsupported tasks which could potentially be of interest to those exploring the data. Using the methodology, the tasks supported and not supported by the visualisation system were revealed.

The task classification may also be of benefit in the other evaluation scenarios outlined in Section 2.3.5, such as when selecting representative tasks for use in experiments by presenting the range of possible tasks for inclusion in evaluation and offering justification for selected tasks.

Finally, the method outlined for constructing the task design space (based on referential components) could potentially be applied when constructing design spaces for other types of data with two referrers, such as spatio-temporal data.

(2) A characterisation of temporal graph visualisation techniques

Based on existing classifications of graph and temporal graph techniques, and also classifications from related areas, two independent dimensions (time and graph structural encodings) were identified and used to construct a design space for temporal graph visualisation techniques (Section 7.1). Existing techniques from the literature were then mapped to this design space (Section 7.3). This mapping not only brings order to the array of temporal graph techniques proposed in the literature (particularly useful to researchers new to the area), but also reveals a number of less explored and unexplored possibilities; these may prove fruitful avenues for researchers interested in developing novel techniques.

(3) A review of techniques to support temporal graph tasks, revealing less well supported and unsupported tasks.

The techniques to support each of the identified task categories were reviewed. As the tasks are domain independent, techniques from a across a wide range of domains could be considered. Interestingly, because the taxonomy is extended from a generic framework, the techniques required to support the tasks were found to be much wider than those which most temporal graph visualisation techniques currently consider. This highlighted the need for the inclusion of techniques from general, temporal and static graph visualisation research areas when developing systems.

The task-technique mapping also draws attention to an important distinction which had been overlooked in the literature: that nested views are able to represent the A3 and S3 aspectual behaviours (distributions of temporal trends over the graph), while the other types of visual approaches for temporal graphs represent graph evolution over time (A4 and S4 behaviours). To date, any technique used in conjunction with temporal graph data has simply been considered to be a “temporal graph visualisation technique”, regardless of which aspects of the data it is able to show.

Several areas in which further research is needed are highlighted by the mapping. Firstly, additional empirical studies are needed to evaluate the performance of different graph and temporal encodings. This would help establish which encodings are most appropriate in which data scenarios. Secondly, little attention has been paid to comparison and relation seeking tasks in Q4. We can think of many real world analysis scenarios in which such techniques could be beneficial, for example, comparing the spread of disease in a public health network before and after an intervention; comparing different types of communication networks (phone, face-face, email) over the same time period; comparing the changes in organisational structure under different management or across different companies. This area of research is likely to have a number of interesting challenges associated with it, not least for developing appropriate layout algorithms to support a person’s understanding of two or more simultaneously evolving graphs. Note that these unsupported tasks would not have been identified had extant task classifications been used as the basis of the mapping.

The mapping also highlighted that understanding of attribute values in temporal graphs is less well considered than visualising the structural aspects of the data. This is understandable as graph structure is often the main aspect of the data which tools seek to represent, however, the vast majority of real world analysis scenarios are likely to involve some sort of attribute value, either associated with the nodes or edges. Further work in this area would therefore be well justified.

The variety of comparison tasks identified in the framework point to the need to develop systems which allow flexible selection of temporal graph components for use in comparison views. More generally, further work is needed in integrating tools and views of the data in order to support the wide variety of tasks identified in the framework, which require support from very different visual techniques. Finally, further research could usefully be directed toward tools to support synthesis of findings, and recording analysis histories.

The usefulness of the mapping was demonstrated in the case study of Chapter 9, where it was possible to make recommendations for potential techniques to add to the existing system.

(4) A review of classification construction and evaluation practices

Despite the recent interest amongst the visualisation community in design and evaluation practices for developing visualisation systems and tools, the design and evaluation processes involved in developing frameworks such as task classifications to help support these endeavours has received very little attention to date. The final contribution of this thesis therefore is its elucidation of the task classification construction process, the threats to validity at each stage of construction and means of mitigating these threats, along with detailed consideration of the appropriateness of evaluation strategies according to the different aspects of the classification which they seek to evaluate. These discussions were presented in Sections 3.1 and 6.1 respectively. It is hoped that the guidance arising from these investigations will be of benefit to developers of classifications in determining appropriate construction and evaluation strategies, and also be of use to those selecting between competing classifications for use in the design and evaluation processes.

10.3 Future work

Let us now consider the opportunities for future work identified in this thesis.

10.3.1 Identifying ‘real world’ tasks

One limitation of using a formal approach to specify tasks is that it does not provide information as to whether a task is a ‘real world’ task (i.e. one which people will find helpful to carry out), or simply a construct of the formal process used to construct the classification.

The evaluations of Chapter 6 examined only part of the task framework with respect to the real world nature of tasks. At a high level, evidence drawn from extant frameworks, tasks of interest in the empirical study, and examples from the literature, supported the real world nature of the main task types (lookup, comparison, relation seeking) in each of the quadrants, and the connection discovery tasks. Further work at a finer level of granularity, to cover inverse and direct variations of tasks, along with the further dimensions which were not explored in the evaluations (e.g. the extent to which data items are specified in tasks, or same/different time/graph components are involved in comparison tasks etc.) is also needed.

As noted in Section 6.1.3.3, evaluating the ‘real world’ nature of tasks can prove difficult. For a large domain independent classification such as is outlined in this thesis, examples may need to be sought from multiple domains to cover all possible task variations, which may require input from multiple domain experts. Even where we fail to find an example of a task, it cannot be concluded that the category is redundant; it could simply be a task of interest in a rather niche analysis scenario or particular to a specific domain.

One important point to note – as outlined in the discussion of Chapter 8 - relates to the gaps in task support identified in the task-technique mapping. Where a task is identified as an opportunity for further research as it is not currently well supported, before developing techniques, we should be sure to consider the evidence in support of whether it is real world in nature, either as identified in this thesis, or from further consideration of potential domains and analysis scenarios.

10.3.2 Evaluating the usability of the task framework

As noted in Chapter 6, one aspect of the task framework which remains to be evaluated as its usability by other visualisation researchers. While it has been shown to be more comprehensive than extant frameworks, and useful in both the design and evaluation processes, further work needs to be carried out in order to establish how easy it is for other visualisation researchers to utilise the framework. The adoption (or otherwise) of the published version of the task framework may be one indicator as to its usability.

10.3.3 Improving evaluation

One of the potential uses of a task-technique mapping is in identifying opportunities for evaluations. In the case where tasks are supported by multiple techniques, the question arises as to which offers better support. Most of the evaluations in the temporal graph literature to date have focussed on assessing the effect of layout stability on the mental map [131], and comparing sequential and juxtaposed temporal encodings [18], [78]. The task-technique mapping revealed several areas in which multiple techniques were available to support the same task (not least the mapping of the entire temporal graph visualisation design space to the Q4 behaviour characterisation task), thus potentially yielding interesting opportunities for evaluation.

An interesting finding made during the literature review was the surprisingly limited discussion of tasks in both the systems and techniques literature, and in papers describing controlled studies. This suggests that there is room for further work in evaluating existing techniques using a wider range of tasks than have previously been considered.

10.3.4 *Developing novel techniques*

The mapping of existing techniques to the technique design space revealed a number of opportunities for novel combinations of time and graph encodings, which could prove interesting avenues for further research.

Examining which visual techniques could potentially support the task categories of the taxonomy uncovered a number of areas which could benefit from further research. For example, investigating techniques to support comparison in Q4 could be a particularly interesting direction. As discussed in Section 8.8, the real world nature of these tasks should be considered before novel techniques are developed.

10.4 **Summary**

Temporal graph visualisation is an upcoming and important area of Information Visualisation, being of relevance across a wide range of domains and application areas. This thesis offers three main tools – a comprehensive task framework, a visualisation technique design space, and a task-technique mapping - which are intended to be of assistance to those researching in the area of temporal graph visualisation. They have been developed to help support communication amongst researchers in the area; bring order to the work that has been carried out to date; reveal opportunities for future work; and offer assistance in the design and evaluation processes.

The work has investigated the tasks involved in exploring temporal graph data and the visual techniques for their support. It has provided a domain-independent taxonomy and design space of temporal graph tasks, which was shown to have greater task coverage than extant taxonomies, and demonstrated to be of use in both the design and evaluation processes. A design space of possible visual encodings for representing temporal graph data was constructed, and a mapping of the existing techniques in the literature brings order to the work that has been carried out to date, and revealed a number of interesting unexplored possibilities for representing this type of data. Mapping the visual techniques which support the different task types exposed the very wide range of techniques – spanning multiple research areas – which are required to fully support exploration of temporal graph data. It also revealed a number of areas in which further research is required in order to develop techniques to support the full range of temporal graph tasks.

References

[1] S. K. Card, J. D. Mackinlay, and B. Shneiderman, *Readings in information visualization : using vision to think*. San Francisco, Calif.: Morgan Kaufmann Publishers, 1999.

[2] F. Beck, M. Burch, S. Diehl, and and D. Weiskopf, "Digital Library for publication The State of the Art in Visualizing Dynamic Graphs," 2014. [Online]. Available: http://dynamicgraphs.fbeck.com/.

[3] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski, "Visualizing time-oriented data—A systematic view," *Comput. Graph.*, vol. 31, no. 3, pp. 401–409, Jun. 2007.

[4] J. Fekete, J. J. Van Wijk, J. T. Stasko, and C. North, "The Value of Information Visualization," *Inf. Vis.*, vol. 4950, no. 2, pp. 1–18, 2008.

[5] N. Andrienko and G. Andrienko, *Exploratory analysis of spatial and temporal data: a systematic approach*. New York: Springer, 2006.

[6] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical studies in information visualization: Seven scenarios," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 9, pp. 1520–1536, 2012.

[7] E. R. A. Valiati, M. S. Pimenta, and C. M. D. S. Freitas, "A taxonomy of tasks for guiding the evaluation of multidimensional visualizations," in *Proceedings of the 2006 AVI workshop on BEyond time and errors novel evaLuation methods for Information Visualization - BELIV '06*, 2006.

[8] S. Wehrend and C. Lewis, "A problem-oriented classification of visualization techniques," *Proc. First IEEE Conf. Vis. Vis. `90*, p. 139–143, 1990.

[9] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Moller, "A systematic review on the practice of evaluating visualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2818–2827, 2013.

[10] T. Munzner, "A Nested Model for Visualization Design and Validation," *Vis. Comput. Graph. IEEE Trans.*, vol. 15, no. 6, pp. 921–928, 2009.

[11] M. Meyer, M. Sedlmair, and T. Munzner, "The Four-Level Nested Model Revisited : Blocks and Guidelines," 2012.

[12] M. Sedlmair, M. Meyer, and T. Munzner, "Design Study Methodology : Reflections from the Trenches and the Stacks," no. October, 2012.

[13] J. S. Yi, Y. A. Kang, J. Stasko, and J. Jacko, "Toward a deeper understanding of the role of interaction in information visualization.," *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1224–31, 2007.

[14] C. Bennett, J. Ryall, L. Spalteholz, and A. Gooch, "The Aesthetics of Graph Visualization," *Comput. Aesthet. Graph. Vis. Imaging*, pp. 57–64, 2007.

[15] T. Von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. Van Wijk, J. Fekete, and D. Fellner, "Visual Analysis of Large Graphs," *Proc. Eurographics State Art Rep.*, vol. 2, no. 7, Dec. 2010.

[16] D. Archambault and H. C. Purchase, "The mental map and memorability in dynamic graphs," *2012 IEEE Pacific Vis. Symp.*, pp. 89–96, Feb. 2012.

[17] D. Archambault, H. C. Purchase, and B. Pinaud, "Difference Map Readability for Dynamic Graphs," in *18th International Symposium on Graph Drawing*, 2010, vol. 6502, pp. 50–61.

[18] D. Archambault, H. Purchase, and B. Pinaud, "Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs.," *IEEE Trans. Vis. Comput. Graph.*, pp. 1–14, May 2010.

[19] H. C. Purchase, E. Hoggan, and C. Gorg, "How Important is the ' Mental Map '? – an Empirical Investigation of a Dynamic Graph Layout Algorithm," in *Proceedings of 14th International Symposium on Graph Drawing 2006*, 2006.

[20] H. Purchase and A. Samra, "Extremes Are Better: Investigating Mental Map Preservation in Dynamic Graphs," in *Diagrammatic Representation and Inference*, G. Stapleton, J. Howse, and J. Lee, Eds. Berlin / Heidelberg: Springer, 2008, pp. 60–73.

[21] P. Saffrey and H. Purchase, "The ' Mental Map ' versus ' Static Aesthetic ' Compromise in Dynamic Graphs : A User Study," in *Australian User Interface Conference (AUIC)*, 2008, vol. 76, no. January, pp. 85–94.

[22] S. Diehl, C. Goerg, and A. Kerren, "Preserving the mental map using foresighted layout.," in *Proceedings of Joint Eurographics, IEEE TCVG Symposium on Visualization (Ascona, Switzerland, 2001)*, 2001, pp. 175–184.

[23] C. Erten, P. Harding, S. Kobourov, and K. Wampler, "GraphAEL : Graph Animations with Evolving Layouts," *Graph Draw. Lect. Notes Comput. Sci.*, vol. 2912/2004, pp. 98–110, 2004.

[24] G. Kumar and M. Garland, "Visual exploration of complex time-varying graphs.," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 805–12, 2006.

[25] Y. Frishman and A. Tal, "Online dynamic graph drawing.," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 4, pp. 727–40, 2008.

[26] K. S. Xu, M. Kliger, and A. O. Hero III, "Visualizing the Temporal Evolution of Dynamic Networks," in *Proc. 9th Workshop on Mining and Learning with Graphs*, 2011.

[27] D. Archambault, H. Purchase, and B. Pinaud, "Animation, Small Multiples, and the Effect of Mental Map Preservation in Dynamic Graphs.," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 4, pp. 539–552, May 2011.

[28] P. Federico, W. Aigner, S. Miksch, F. Windhager, and L. Zenk, "A Visual Analytics Approach to Dynamic Social Networks," in *Proceedings of the 11th International Conference on Knowledge Management and Knowledge Technologies (i-KNOW '11)*, 2011.

[29] J. Moody, D. McFarland, and S. Bender-deMoll, "Dynamic Network Visualization," *Am. J. Sociol.*, vol. 110, no. 4, pp. 1206–1241, Jan. 2005.

[30] K. D. Bailey, *Typologies and taxonomies: an introduction to classification techniques*, no. 7. Sage, 1994.

[31] A. J. Cain, “Taxonomy,” *Encyclopædia Britannica*, 2011. [Online]. Available: https://www.britannica.com/science/taxonomy/The-Linnaean-system. [Accessed: 01-Feb-2016].

[32] R. Smith, K. Rassmann, H. Davies, and N. King, “Why Taxonomy Matters,” Egham, UK, 2011.

[33] T. E. Moffitt, “Adolescence-limited and life-course-persistent antisocial behavior: a developmental taxonomy.,” *Psychol. Rev.*, vol. 100, no. 4, p. 674, 1993.

[34] K. J. Gershenson and A. L. Stauffer, “A Taxonomy for Design Requirements from Corporate Customers,” *Res. Eng. Des.*, vol. 11, no. 2, pp. 103–115, 1999.

[35] B. S. Bloom, M. D. Engelhart, E. J. Furst, W. H. Hill, and D. R. Krathwohl, *Taxonomy qf Educational Objectives: The Classification of Educational Goals. Handbook I: Cognitive Domain*. New York: David McKay Company., 1956.

[36] R. Amar, J. Eagan, and J. Stasko, “Low-level components of analytic activity in information visualization,” *Proc. - IEEE Symp. Inf. Vis. INFO VIS*, pp. 111–117, 2005.

[37] H. J. Schulz, T. Nocke, M. Heitzler, and H. Schumann, “A design space of visualization tasks,” *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2366–2375, 2013.

[38] H.-J. Schulz, S. Hadlak, and H. Schumann, “The Design Space of Implicit Hierarchy Visualization: A Survey.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 4, pp. 393–411, May 2011.

[39] W. Javed and N. Elmqvist, “Exploring the design space of composite visualization,” *2012 IEEE Pacific Vis. Symp.*, pp. 1–8, Feb. 2012.

[40] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry, “Task taxonomy for graph visualization,” in *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, 2006, pp. 81–85.

[41] J. Ahn, C. Plaisant, and B. Shneiderman, “A Task Taxonomy for Network Evolution Analysis,” *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 3, pp. 365–376, 2014.

[42] S. Wehrend and C. Lewis, “A problem-oriented classification of visualization techniques,” in *Proceedings of the 1st conference on Visualization ’90 (VIS ’90)*, 1990, pp. 139–143.

[43] A. Rind, W. Aigner, M. Wagner, S. Miksch, and T. Lammarsch, “Task cube: A three-dimensional conceptual space of user tasks in visualization design and evaluation,” *Inf. Vis.*, 2015.

[44] D. Sacha, A. Stoffel, F. Stoffel, B. Kwon, G. Ellis, and D. Keim, “Knowledge Generation Model for Visual Analytics,” *Vis. Comput. Graph. IEEE Trans.*, vol. 20, no. 12, pp. 1604–1613, 2014.

[45] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann, "A design space of visualization tasks.," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2366–75, Dec. 2013.

[46] R. A. Amar and J. T. Stasko, "Knowledge precepts for design and evaluation of information visualizations," *IEEE Trans. Vis. Comput. Graph.*, vol. 11, no. 4, pp. 432–442, 2005.

[47] K. Sedig and P. Parsons, "Interaction design for complex cognitive activities with visual representations: A pattern-based approach," *AIS Trans. Human-Computer Interact.*, vol. 5, no. 1, pp. 84–133, 2013.

[48] J. Heer and B. Shneiderman, "Interactive Dynamics for Visual Analysis," *Queue - Micoprocessors*, vol. 10, no. 2, p. 30, 2012.

[49] T. Lammarsch, A. Rind, W. Aigner, and S. Miksch, "Developing an Extended Task Framework for Exploratory Data Analysis Along the Structure of Time," in *EuroVA 2012: International Workshop on Visual Analytics*, 2012, pp. 31–35.

[50] S. McKenna, D. Mazur, J. Agutter, and M. Meyer, "Design Activity Framework for Visualization Design," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2191–2200, Dec. 2014.

[51] C. Plaisant, "The challenge of information visualization evaluation," *Proc. Work. Conf. Adv. Vis. interfaces - AVI '04*, pp. 109–116, 2004.

[52] J. J. Van Wijk, "Bridging the Gaps," *IEEE Comput. Graph. Appl.*, vol. 26, no. 6, pp. 6–9, 2006.

[53] M. Brehmer, M. Sedlmair, S. Ingram, and T. Munzner, "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences," in *In ACM Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization.*, 2014, pp. 1–8.

[54] T. Munzner, *Visualization Analysis and Design*. CRC Press, 2014.

[55] M. Tory and T. Moller, "Rethinking Visualization: A High-Level Taxonomy," *IEEE Symp. Inf. Vis.*, pp. 151–158, 2004.

[56] B. Shneiderman, "The eyes have it: a task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, 1996, pp. 336–343.

[57] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo.," *Commun. ACM*, pp. 59–67, 2010.

[58] H. Ltifi, M. Ben Ayed, A. M. Alimi, and S. Lepreux, "Survey of information visualization techniques for exploitation in KDD," in *IEEE/ACS International Conference on Computer Systems and Applications, 2009. AICCSA 2009.*, 2009, no. 1, pp. 218–225.

[59] J. Heer and B. Shneiderman, "Interactive Dynamics for Visual Analysis: A taxonomy of tools that support the fluent and flexible use of visualizations," *Commun. ACM*, vol. 55, no. 4, pp. 45–54, 2012.

[60] J. I. Maletic, A. Marcus, and M. L. Collard, "A task oriented view of software visualization," *Proc. First Int. Work. Vis. Softw. Underst. Anal.*, pp. 32–40, 2002.

[61] M. C. Chuah and S. F. Roth, "On the semantics of interactive visualizations," in *Proceedings IEEE Symposium on Information Visualization '96*, 1996, pp. 29–36.

[62] R. E. Roth, "An Empirically-Derived Taxonomy of Interaction Primitives for Interactive Cartography and Geovisualization," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2356–2365, 2013.

[63] D. Gotz and M. X. Zhou, "Characterizing Users' Visual Analytic Activity for Insight Provenance," *Inf. Vis.*, vol. 8, no. 1, pp. 42–55, 2009.

[64] T. von Landesberger, S. Fiebig, S. Bremm, A. Kuijper, and D. W. Fellner, "Interaction taxonomy for tracking of user actions in visual analytics applications.," in *Handbook of Human Centric Visualization*, W. Huang, P. Parsons, and K. Sedig, Eds. Springer, 2014, pp. 653–670.

[65] B. Alsallakh, L. Micallef, W. Aigner, H. Hauser, S. Miksch, and P. Rodgers, "Visualizing Sets and Set-typed Data: State-of-the-Art and Future Challenges," *Eurographics Conf. Vis. (EuroVis)– State Art Reports*, pp. 1–21, 2014.

[66] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," *IEEE Symp. Inf. Vis. 2005. INFOVIS 2005.*, pp. 111–117, 2005.

[67] M. Brehmer and T. Munzner, "A multi-level typology of abstract visualization tasks.," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2376–85, Dec. 2013.

[68] A. Rind, W. Aigner, M. Wagner, S. Miksch, and T. Lammarsch, "User Tasks for Evaluation: Untangling the Terminology Throughout Visualization Design and Development," *Proc. Fifth Work. Beyond Time Errors Nov. Eval. Methods Vis.*, pp. 9–15, 2014.

[69] B. Shneiderman and A. Aris, "Network visualization by semantic substrates.," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 733–40, 2006.

[70] J. S. Yi, N. Elmqvist, and S. Lee, "TimeMatrix: Analyzing Temporal Social Networks Using Interactive Matrix-Based Visualizations," *Int. J. Hum. Comput. Interact.*, vol. 26, no. 11–12, pp. 1031–1051, Nov. 2010.

[71] B. Bach, E. Pietriga, and J.-D. Fekete, "GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks.," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 5, pp. 740–754, Nov. 2014.

[72] D. J. Peuquet, "It ' s About Time : A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems," *Ann. Assoc. Am. Geogr.*, vol. 84, no. 3, pp. 441–461, 1994.

[73] P. Saraiya, P. Lee, and C. North, "Visualization of Graphs with Associated Timeseries Data," in *IEEE Symposium on Information Visualization 2005*, 2005, pp. 225–232.

[74] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee, "Exploring the Computing Literature Using Temporal Graph Visualization," in *Proc. SPIE 5295, Visualization and Data Analysis*, 2004.

[75] P. A. Gloor and Y. Zhao, "Analyzing Actors and Their Discussion Topics by

Semantic Social Network Analysis," *Tenth Int. Conf. Inf. Vis.*, pp. 130–135, 2006.

[76] H. Kang, L. Getoor, and L. Singh, "Visual analysis of dynamic group membership in temporal social networks," *ACM SIGKDD Explor. Newsl.*, vol. 9, no. 2, pp. 13–21, Dec. 2007.

[77] M. Ghoniem, J.-D. Fekete, and P. Castagliola, "On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis," *Inf. Vis.*, vol. 4, no. 2, pp. 114–135, May 2005.

[78] M. Farrugia and A. Quigley, "Effective Temporal Graph Layout: A Comparative Study of Animation versus Static Display Methods," *Inf. Vis.*, vol. 10, no. 1, pp. 47–64, 2011.

[79] T. von Landesberger, A. Kuijper, T. Schreck, J. Kohlhammer, J. J. van Wijk, J.-D. Fekete, and D. W. Fellner, "Visual Analysis of Large Graphs: State-of-the-Art and Future Research Challenges," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1719–1749, Apr. 2011.

[80] H.-J. Schulz and H. Schumann, "Visualizing Graphs - A Generalized View," *Tenth Int. Conf. Inf. Vis.*, pp. 166–173, 2006.

[81] M. Graham and J. Kennedy, "A survey of multiple tree visualisation," *Inf. Vis.*, vol. 9, no. 4, pp. 235–252, Nov. 2010.

[82] S. Hadlak, H.-J. Schulz, and H. Schumann, "In Situ Exploration of Large Dynamic Networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2334–2343, 2011.

[83] S. Rufiange and M. J. McGuffin, "DiffAni: visualizing dynamic graphs with a hybrid of difference maps and animation.," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2556–65, Dec. 2013.

[84] F. Beck, M. Burch, S. Diehl, and D. Weiskopf, "The State of the Art in Visualizing Dynamic Graphs," in *Eurographics Conference on Visualization (EuroVis) - STARs*, 2014.

[85] F. Beck, M. Burch, and S. Diehl, "Matching Application Requirements with Dynamic Graph Visualization Profiles," *2013 17th Int. Conf. Inf. Vis.*, vol. 304, pp. 11–18, Jul. 2013.

[86] H. Schumann and W. Müller, *Visualisierung: Grundlagen und allgemeine Methoden*. Springer-Verlag, 2013.

[87] A. M. MacEachren, *How maps work: representation, visualization, and design*. Guilford Press, 1995.

[88] J. J. Thomas and K. A. Cook, Eds., *Illuminating the path:the research and development agenda for visual analytics*. IEEE Computer Society, 2005.

[89] N. Kerracher, J. Kennedy, and K. Chalmers, "A Task Taxonomy for Temporal Graph Visualisation," *Vis. Comput. Graph. IEEE Trans.*, vol. 21, no. 10, pp. 1160–1172, Oct. 2015.

[90] T. Lammarsch, W. Aigner, A. Bertone, S. Miksch, and A. Rind, "Towards a Concept how the Structure of Time can Support the Visual Analytics Process,"

2011.

[91] Z. Liu and J. T. Stasko, “Mental Models , Visual Reasoning and Interaction in Information Visualization : A Top-down Perspective,” *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 999–1008, 2010.

[92] M. Meyer, T. Munzner, and H. Pfister, “MizBee: A multiscale synteny browsers,” *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 897–904, 2009.

[93] A. J. Pretorius, H. C. Purchase, and J. T. Stasko, “Tasks for multivariate network analysis,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8380 LNCS, pp. 77–95, 2014.

[94] A. Rind, T. D. Wang, W. Aigner, S. Miksch, K. Wongsuphasawat, C. Plaisant, and B. Shneiderman, “Interactive Information Visualization to Explore and Query Electronic Health Records,” *Found. Trends Human–Computer Interact.*, vol. 5, no. 3, pp. 207–298, 2013.

[95] X. Suo, “A Task-Centered Visualization Design Environment and a Method for Measuring the Complexity of Visualization Designs,” Georgia State University, Atlanta, 2009.

[96] R. Sakai and J. Aerts, “Card Sorting Techniques for Domain Characterization in Problem-driven Visualization Research,” *Eurographics Conf. Vis.*, 2015.

[97] M. M. K. Hlava, “The Taxobook: Principles and Practices of Building Taxonomies, Part 2 of a 3-Part Series,” *Synth. Lect. Inf. Concepts, Retrieval, Serv.*, vol. 6, no. 4, pp. 1–164, 2014.

[98] W. Aigner, S. Miksch, H. Schumann, and C. Tominski, *Visualization of Time-Oriented Data*. London: Springer, 2011.

[99] J. Bertin, *Semiology of graphics: diagrams, networks, maps.* Madison: University of Wisconsin Press, 1983.

[100] R. Shannon, A. Quigley, and P. Nixon, “Time Sequences,” in *CHI 2009*, 2009, pp. 4615–4620.

[101] S. Asur, S. Parthasarathy, and D. Ucar, “An event-based framework for characterizing the evolutionary behavior of interaction graphs,” *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD ’07*, p. 913, 2007.

[102] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts, “Visual comparison for information visualization,” *Inf. Vis.*, vol. 10, no. 4, pp. 289–309, Sep. 2011.

[103] M. Hascoët and P. Dragicevic, “Interactive graph matching and visual comparison of graphs and clustered graphs,” in *AVI ’12 Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2012, pp. 522–529.

[104] R. Gove, N. Gramsky, R. Kirby, E. Sefer, A. Sopan, C. Dunne, B. Shneiderman, and M. Taieb-Maimon, “NetVisia : Heat Map & Matrix Visualization of Dynamic Social Network Statistics & Content,” in *Proc. IEEE Conference on Social Computing*, 2011, pp. 19–26.

[105] J. Roberts, “A Brief Introduction to Social Network Analysis.” [Online].

Available: http://web.mit.edu/vdb/www/6.977/l-jenn.pdf. [Accessed: 30-Apr-2015].

[106] N. a Christakis and J. H. Fowler, “The spread of obesity in a large social network over 32 years.,” *N. Engl. J. Med.*, vol. 357, no. 4, pp. 370–9, Jul. 2007.

[107] C. North, “Toward Measuring Visualisation Insight,” *Vis. Viewpoints*, vol. 26, no. 3, pp. 6–9, 2006.

[108] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale, “Grounded evaluation of information visualizations,” in *Proceedings of the 2008 Workshop on BEyond time and errors novel evaLuation methods for Information Visualization - BELIV ’08*, 2008, p. 6.

[109] M. Beaudouin-Lafon, “Designing interaction, not interfaces,” in *Proceedings of the working conference on Advanced visual interfaces*, 2004, pp. 15–22.

[110] B. Bach, E. Pietriga, and J.-D. Fekete, “GraphDiaries: Animated Transitions and Temporal Navigation for Dynamic Networks.,” *IEEE Trans. Vis. Comput. Graph.*, pp. 1–14, Nov. 2013.

[111] N. Kerracher, J. Kennedy, and K. Chalmers, “A Task Taxonomy for Temporal Graph Visualisation,” *Trans. Vis. Comput. Graph.*, vol. PP, no. 99, p. 1.1, 2015.

[112] M. Burch and D. Weiskopf, “Visualizing dynamic quantitative data in hierarchies - timeedgetrees: Attaching dynamic weights to tree edges,” in *Proc. Int. Conf. Information Visualization Theory and Applications*, 2011, pp. 177–186.

[113] N. H. Riche, S. Carpendale, T. Madhyastha, N. Roussel, and T. J. Grabowski, “LinkWave: a visual adjacency list for dynamic weighted networks,” in *Proceedings of the 26th Conference on l’Interaction Homme-Machine (IHM ’14)*, 2014, pp. 113–122.

[114] H. Hochheiser and B. Shneiderman, “Dynamic query tools for time series data sets: Timebox widgets for interactive exploration,” *Inf. Vis.*, vol. 3, no. 1, pp. 1–18, 2004.

[115] P. Saraiya, P. Lee, and C. North, “Visualization of graphs with associated timeseries data,” *Proc. - IEEE Symp. Inf. Vis. INFO VIS*, pp. 225–232, 2005.

[116] H. Gibson, J. Faith, and P. Vickers, “A survey of two-dimensional graph layout techniques for information visualisation,” *Inf. Vis.*, vol. 12, no. 3–4, pp. 324–357, 2013.

[117] I. Herman, G. Melancon, and M. S. Marshall, “Graph visualization and navigation in information visualization: A survey,” *Vis. Comput. Graph. IEEE Trans.*, vol. 6, no. 1, pp. 24–43, 2000.

[118] G. a Pavlopoulos, T. G. Soldatos, A. Barbosa-Silva, and R. Schneider, “A reference guide for tree analysis and visualization.,” *BioData Min.*, vol. 3, no. 1, p. 1, Jan. 2010.

[119] W. Aigner, S. Miksch, W. Müller, H. Schumann, and C. Tominski, “Visual methods for analyzing time-oriented data.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 1, pp. 47–60, 2008.

[120] E. Wohlfart, W. Aigner, A. Bertone, and S. Miksch, “Comparing Information Visualization Tools Focusing on the Temporal Dimensions,” *2008 12th Int. Conf. Inf. Vis.*, pp. 69–74, Jul. 2008.

[121] S. F. Silva and T. Catarci, “Visualization of Linear Time-Oriented Data : a Survey,” in *Web Information Systems Engineering, 2000. Proceedings of the First International Conference on*, 2000, no. March, pp. 310–319.

[122] C. Ware, H. Purchase, L. Colpoys, and M. McGill, “Cognitive measurements of graph aesthetics,” *Inf. Vis.*, vol. 1, no. 2, pp. 103–110, Jun. 2002.

[123] H. Schulz, “Treevis.net: A Tree Visualization Reference,” *Comput. Graph. Appl. IEEE*, vol. 31, no. 6, pp. 11–15, 2011.

[124] K.-C. Feng, C. Wang, H.-W. Shen, and T.-Y. Lee, “Coherent Time-Varying Graph Drawing with Multi-Focus+Context Interaction.,” *IEEE Trans. Vis. Comput. Graph.*, pp. 1–12, Jul. 2011.

[125] L. Leydesdorff, T. Schank, A. Scharnhorst, and W. De Nooy, “Animating the development of Social Networks over time using a dynamic extension of multidimensional scaling,” pp. 1–47, 2008.

[126] T. E. Gorochowski, M. di Bernardo, and C. S. Grierson, “Using Aging to Visually Uncover Evolutionary Processes on Networks.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1343–1352, Aug. 2011.

[127] U. Brandes, N. Indlekofer, and M. Mader, “Visualization methods for longitudinal social networks and stochastic actor-oriented modeling,” *Soc. Networks*, Jul. 2011.

[128] P. Federico, J. Pfeffer, W. Aigner, S. Miksch, and L. Zenk, “Visual Analysis of Dynamic Networks using Change Centrality,” in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 2012.

[129] D. Eppstein, Z. Galil, and G. Italiano, “Dynamic Graph Algorithms,” in *CRC Handbook of Algorithms and Theory of Computation*, vol. 20244, no. 20244, M. J. Atallah, Ed. CRC Press, 1999, p. chapter 8.

[130] C. Friedrich and M. Houle, “Graph Drawing in Motion II,” in *GRAPH DRAWING Lecture Notes in Computer Science*, P. Mutzel, M. Jünger, and S. Leipert, Eds. Berlin / Heidelberg: Springer, 2002, pp. 220–231.

[131] D. Archambault and H. C. Purchase, “The ‘ Map ’ in the Mental Map : Experimental Results in Dynamic Graph Drawing,” *Int. J. Hum. Comput. Stud.*, vol. 71, no. 11, pp. 1044–1055, 2013.

[132] D. Archambault and H. C. Purchase, “Mental Map Preservation Helps User Orientation in Dynamic Graphs,” *GRAPH Draw. Lect. Notes Comput. Sci.*, vol. 7704, pp. 475–486, 2013.

[133] C. McGrath and J. Blythe, “Do you see what i want you to see? the effects of motion and spatial layout on viewers’ perceptions of graph structure,” *J. Soc. Struct.*, vol. 5, no. 2, 2004.

[134] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press, 1983.

[135] M. Burch and S. Diehl, “TimeRadarTrees: Visualizing Dynamic Compound Digraphs,” *Comput. Graph. Forum*, vol. 27, no. 3, pp. 823–830, May 2008.

[136] M. Farrugia, N. Hurley, and A. Quigley, “Exploring temporal ego networks using small multiples and tree-ring layouts,” in *ACHI 2011, The Fourth International Conference on Advances in Computer-Human Interactions*, 2011, pp. 79–88.

[137] M. Rosvall and C. T. Bergstrom, “Mapping change in large networks.,” *PLoS One*, vol. 5, no. 1, p. e8694, Jan. 2010.

[138] A. Ahmed, T. Dwyer, C. Murray, L. Song, and Y. X. Wu, “InfoVis 2004 Contest : WilmaScope Graph Visualisation,” in *INFOVIS 2004, IEEE Symposium on*, 2004.

[139] U. Brandes and S. R. Corman, “Visual Unrolling of Network Evolution and the Analysis of Dynamic Discourse,” *Inf. Vis.*, vol. 2, no. 1, pp. 40–50, 2003.

[140] L. Shi, C. Wang, and Z. Wen, “Dynamic Network Visualization in 1 . 5D,” in *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, 2011, pp. 179–186.

[141] C. Erten, P. J. Harding, S. G. Kobourov, K. Wampler, and G. Yee, “Exploring the Computing Literature Using Temporal Graph Visualization,” *Vis. Data Anal. Proc. SPIE*, vol. 5295, pp. 45–56, Jun. 2004.

[142] S. Everton, “A guide for the visually perplexed: visually representing social networks,” Stanford University, 2004.

[143] C. Collins and S. Carpendale, “VisLink: revealing relationships amongst visualizations.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1192–9, 2006.

[144] D. Archambault, “Structural Differences Between Two Graphs through Hierarchies,” in *Graphics Interface*, 2009, pp. 87–94.

[145] L. Zaman, A. Kalra, and W. Stuerzlinger, “The Effect of Animation , Dual View , Difference Layers , and Relative Re-Layout in Hierarchical Diagram Differencing,” in *Proceedings of Graphics Interface 2011*, 2011, pp. 183–190.

[146] Y. Tu and H.-W. Shen, “Visualizing changes of hierarchical data using treemaps.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1286–93, 2007.

[147] Smithsonian Institution, “HistoryWired: A few of our favorite things.” [Online]. Available: http://historywired.si.edu/.

[148] D. Turo and B. Johnson, “Improving the visualization of hierarchies with treemaps: design issues and experimentation,” *Proc. Vis. ’92*, pp. 124–131, 1992.

[149] M. Ghoniem and J. Fekete, “Animating Treemaps,” in *Proc. of 18th HCIL Symposium-Workshop on Treemap Implementations and Applications.*, 2001.

[150] T. Tekusova and T. Schreck, “Visualizing Time-Dependent Data in Multivariate Hierarchic Plots - Design and Evaluation of an Economic Application,” in *IV08 Proceedings of theInternational Conference on Information Visualisation*, 2008, pp. 143–150.

[151] K. Russel and S. Carter, “How the Giants of Finance Shrank, Then Grew, Under the Financial Crisis,” *New York Times*, 2009. [Online]. Available: http://www.nytimes.com/interactive/2009/09/12/business/financial-

markets-graphic.html?_r=2&.

[152] D. Brvar, A. Mrvar, and V. Batagelj, “Pajek to SVG Anim - Visualization of temporal networks in SVG.” [Online]. Available: http://pajek.imfm.si/doku.php?id=faq:svganim.

[153] U. Brandes and D. Wagner, “Visone: Visual Social Networks.” [Online]. Available: http://visone.info/.

[154] C. Chen and S. Morris, “Visualizing evolving networks: minimum spanning trees versus pathfinder networks,” *IEEE Symp. Inf. Vis. 2003 (IEEE Cat. No.03TH8714)*, pp. 67–74, 2003.

[155] P. A. Gloor, R. Laubacher, and S. B. C. Dynes, “Visualization of Communication Patterns in Collaborative Innovation Networks - Analysis of Some W3C Working Groups,” pp. 56–60, 2003.

[156] P. A. Gloor, R. Laubacher, Y. Zhao, and S. B. C. Dynes, “Temporal Visualization and Analysis of Social Networks,” in *NAACSOS Conference, June 27 - 29*, 2004.

[157] P. A. Gloor and Y. Zhao, “TeCFlow – A Temporal Communication Flow Visualizer for Social Network Analysis,” in *CSCW’04 Workshop on Social Networks*, 2004.

[158] P. Mutton, “Inferring and visualizing social networks on internet relay chat,” in *Proceedings of the Eigth International Conference on Information Visualisation (IV ’04)*, 2004.

[159] S. Bender-deMoll and D. A. Mcfarland, “The Art and Science of Dynamic Network Visualization,” *JoSS*, vol. 7, no. 2, 2006.

[160] S. K. Card, B. Suh, B. A. Pendleton, J. Heer, and J. W. Bodnar, “TimeTree : Exploring Time Changing Hierarchies,” in *Proceedings on the Symposium on Visual Analytics Science and Technology*, 2006, pp. 3–10.

[161] A. Perer and B. Shneiderman, “Balancing systematic and flexible exploration of social networks.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 693–700, 2006.

[162] M. Trier and A. Bobrik, “Analyzing the dynamics of community formation using brokering activities,” *Communities Technol. 2007*, pp. 463–477, 2007.

[163] C. Correa, T. Crnovrsanin, C. Meulder, Z. Shen, R. Armstrong, J. Shearer, and K.-L. Ma, “Visual analytics of cell phone data using MobiVis and OntoVis,” in *Visual Analytics Science and Technology, 2008. VAST ’08. IEEE Symposium on*, 2008, pp. 211–212.

[164] W. de Nooy, “Signs Over Time : Statistical and Visual Analysis of a Longitudinal Signed Network,” *J. Soc. Struct.*, vol. 9, no. 1, 2008.

[165] M. Farrugia and A. Quigley, “Cell phone Mini Challenge: Node-link animation award animating multivariate dynamic social networks,” *2008 IEEE Symp. Vis. Anal. Sci. Technol.*, pp. 215–216, Oct. 2008.

[166] E. Loubier and B. Dousset, “Temporal and relational data representation by graph morphing,” in *Safety and Reliability for managing Risk (ESREL 2008)*, 2008.

[167] M. Pohl, F. Reitz, and P. Birke, “As Time Goes by – Integrated Visualization and

Analysis of Dynamic Networks," *Proc. Work. Conf. Adv. Vis. interfaces (AVI '08)*, pp. 372–375, 2008.

[168] E. Swing, "Award: Efficient toolkit integration solving the cell phone calls challenge with the Prajna Project," *2008 IEEE Symp. Vis. Anal. Sci. Technol.*, pp. 221–222, Oct. 2008.

[169] M. Trier, "Towards Dynamic Visualization for Understanding Evolution of Digital Communication Networks," *Inf. Syst. Res.*, vol. 19, no. 3, pp. 335–350, 2008.

[170] M. Bastian, S. Heymann, and M. Jacomy, "Gephi : An Open Source Software for Exploring and Manipulating Networks," in *International AAAI Conference on Weblogs and Social Media.*, 2009, pp. 361–362.

[171] D. Chang, Y. Ge, S. Song, N. Coleman, J. Christensen, and J. Heer, "Visualizing the Republic of Letters." Stanford University, Staford, 2009.

[172] J. Ahn, M. Taieb-maimon, and A. Sopan, "Temporal Visualization of Social Network Dynamics : Prototypes for Nation of Neighbors," *Proc. Soc. Comput. Behav. Model. Predict. Conf.*, pp. 309–316, 2010.

[173] M. Itoh, M. Toyoda, and M. Kitsuregawa, "An Interactive Visualization Framework for Time-Series of Web Graphs in a 3D Environment," *2010 14th Int. Conf. Inf. Vis.*, no. vi, pp. 54–60, Jul. 2010.

[174] R. Shannon, A. Quigley, and P. Nixon, "Showtime : Increasing viewer understanding of dynamic network visualisations," in *Proceedings of Advanced Visual Interfaces, AVI2010*, 2010.

[175] R. E. Curtis, A. Yuen, L. Song, A. Goyal, and E. P. Xing, "TVNViewer : An interactive visualization tool for exploring net- works that change over time or space," *Bioinformatics*, vol. 10, pp. 1–2, 2011.

[176] K. S. Xu, M. Kliger, and A. O. H. Iii, "Visualizing the Temporal Evolution of Dynamic Networks - Poster," in *Proc. 9th Workshop on Mining and Learning with Graphs*, 2011, vol. 106, no. 36.

[177] I. Boyandin, E. Bertini, and D. Lalanne, "A Qualitative Study on the Exploration of Temporal Changes in Flow Maps with Animation and Small-Multiples," *Comput. Graph. Forum*, vol. 31, no. 3, pp. 1005–1014, 2012.

[178] M. Burch, F. Beck, and D. Weiskopf, "Radial Edge Splatting for Visualizing Dynamic Directed Graphs," in *GRAPP/IVAPP 2012*, 2012, pp. 603–612.

[179] B. Bach, E. Pietriga, and J.-D. Fekete., "Temporal Navigation in Dynamic Networks," in *IEEE Vis Week. 2012.*, 2012.

[180] D. Filippova, M. Fitzgerald, C. Kingsford, and F. Benadon, "Dynamic Exploration of Recording Sessions between Jazz Musicians over Time," *2012 Int. Conf. Privacy, Secur. Risk Trust 2012 Int. Confernece Soc. Comput.*, pp. 368–376, Sep. 2012.

[181] S. Ghani, N. Elmqvist, and J. S. Yi, "Perception of Animated Node-Link Diagrams for Dynamic Graphs," vol. 31, no. 3, 2012.

[182] M. Itoh and M. Akaishi, "Visualization for Changes in Relationships between

Historical Figures in Chronicles," *2012 16th Int. Conf. Inf. Vis.*, vol. 1, no. Figure 1, pp. 283–290, Jul. 2012.

[183] R. Kleeb, P. A. Gloor, K. Nemoto, and M. Henninger, "Wikimaps : dynamic maps of knowledge," *Int. J. Organ. Des. Eng.*, vol. 2, no. 2, pp. 204–224, 2012.

[184] C. Hurter, O. Ersoy, S. I. Fabrikant, T. R. Klein, and A. C. Telea, "Bundled Visualization of Dynamic Graph and Trail Data.," *IEEE Trans. Vis. Comput. Graph.*, Oct. 2013.

[185] Y. Frishman and A. Tal, "Dynamic Drawing of Clustered Graphs," *IEEE Symp. Inf. Vis.*, pp. 191–198, 2004.

[186] D. Holten, B. Cornelissen, and J. J. Van Wijk, "Trace visualization using hierarchical edge bundles and massive sequence views," in *Proc. 4th IEEE Int. Workshop on Visualising Software for Understanding and Analysis*, 2007, pp. 47–54.

[187] F. Reitz, M. Pohl, and S. Diehl, "Focused Animation of Dynamic Compound Graphs," *Inf. Vis. 2009 13th Int. Conf.*, pp. 679–684, Jul. 2009.

[188] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid, "Cerebral: visualizing multiple experimental conditions on a graph with biological context.," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1253–60, 2008.

[189] K. Durant, A. Mccray, and C. Safran, "Modeling the Temporal Evolution of an Online Cancer Forum," *Proc. 1st ACM Int. Heal. INformatics Symp. (IHI 2010)*, pp. 356–365, 2010.

[190] M. Burch, C. Vehlow, F. Beck, S. Diehl, and D. Weiskopf, "Parallel edge splatting for scalable dynamic graph visualization.," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2344–53, Dec. 2011.

[191] F. Beck, M. Burch, C. Vehlow, S. Diehl, and D. Weiskopf, "Rapid Serial Visual Presentation in dynamic graph visualization," *2012 IEEE Symp. Vis. Lang. Human-Centric Comput.*, pp. 185–192, Sep. 2012.

[192] P. Federico, W. Aigner, S. Miksch, M. Smuc, and F. Windhager, "Vertigo zoom : combining relational and temporal perspectives on dynamic networks," in *AVI'12*, 2012.

[193] S. van den Elzen, D. Holten, J. Blaas, and J. J. van Wijk, "Dynamic Network Visualization with Extended Massive Sequence Views.," *IEEE Trans. Vis. Comput. Graph.*, no. 99, pp. 1–1, Nov. 2013.

[194] F. Windhager, M. Smuc, L. Zenk, P. Federico, J. Pfeffer, and W. Aigner, "On Visualizing Knowledge Flows at a University Department," *Procedia - Soc. Behav. Sci.*, vol. 100, pp. 127–143, Nov. 2013.

[195] A. G. Melville, M. Graham, and J. B. Kennedy, "Combined vs. Separate Views in Matrix-based Graph Analysis and Comparison," in *2011 15th International Conference on Information Visualisation*, 2011, pp. 53–58.

[196] A. Perer and J. Sun, "MatrixFlow : Temporal Network Visual Analytics to Track Symptom Evolution during Disease Progression," in *AMIA Annual Symposium Proceedings.*, 2012, p. 716.

[197] B. Bach, E. Pietriga, and J. Fekete, "Visualizing Dynamic Networks with Matrix Cubes," in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems (CHI '14)*, 2014, pp. 877–886.

[198] M. Greilich, M. Burch, and S. Diehl, "Visualizing the Evolution of Compound Digraphs with TimeArcTrees," *Eurographics/ IEEE-VGTC Symp. Vis. 2009*, vol. 28, no. 3, pp. 975–982, 2009.

[199] A. Ahmed, X. Fu, S.-H. Hong, Q. H. Nguyen, and K. Xu, "Visual Analysis of Dynamic Networks with Geological Clustering," *2007 IEEE Symp. Vis. Anal. Sci. Technol.*, pp. 221–222, Oct. 2007.

[200] A. Perer, "Using SocialAction to Uncover Structure in Social Networks over Time," in *IEEE Symposium on Visual Analytics Science and Technology*, 2008, pp. 213–214.

[201] M. E. Joorabchi, J.-D. Yim, and C. D. Shaw, "EmailTime: Visual analytics of emails," *2010 IEEE Symp. Vis. Anal. Sci. Technol.*, pp. 233–234, Oct. 2010.

[202] A. Telea and D. Auber, "Code Flows: Visualizing Structural Evolution of Source Code," *Comput. Graph. Forum*, vol. 27, no. 3, pp. 831–838, May 2008.

[203] K. Reda, C. Tantipathananandh, A. Johnson, J. Leigh, and T. Berger-wolf, "Visualizing the Evolution of Community Structures in Dynamic Social Networks," *Eurographics/IEEE Symp. Vis. 2011 (EuroVis 2011)*, 2011.

[204] T. Dwyer and P. Eades, "Visualising a Fund Manager Flow Graph with Columns and Worms," in *Proceedings of the Sixth International Conference on Information Visualisation (IV'02)*, 2002, pp. 147–152.

[205] G. Groh, H. Hanstein, and W. Wordnl, "Interactively Visualizing Dynamic Social Networks with DySoN," in *IUI'09 Workshop on Visual Interfaces to the Social and the Semantic Web*, 2009.

[206] P. Bak, F. Mansmann, H. Janetzko, and D. a Keim, "Spatiotemporal analysis of sensor logs using growth ring maps.," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 913–20, 2009.

[207] U. Khurana, V. Nguyen, H. Cheng, J. Ahn, X. S. Chen, and B. Shneiderman, "Visual Analysis of Temporal Trends in Social Networks Using Edge Color Coding and Metric Timelines," in *Proc. IEEE Conference on Social Computing 2011*, 2011.

[208] M. C. Hao, U. Dayal, D. a. Keim, and T. Schreck, "Importance-driven visualization layouts for large time series data," *IEEE Symp. Inf. Vis. 2005. INFOVIS 2005.*, pp. 203–210, 2005.

[209] T. Schreck, D. Keim, and F. Mansmann, "Regular TreeMap Layouts for Visual Analysis of Hierarchical Data," in *Spring Conference on Computer Graphics (SCCG'2006)*, 2005.

[210] K. Stein, R. Wegener, and C. Schlieder, "Pixel-Oriented Visualization of Change in Social Networks," *2010 Int. Conf. Adv. Soc. Networks Anal. Min.*, pp. 233–240, Aug. 2010.

[211] U. Brandes and B. Nick, "Asymmetric relations in longitudinal social networks.," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2283–90, Dec.

2011.

[212] K. Thiel, F. Dill, T. Kötter, and M. Berthold, “Towards visual exploration of topic shifts,” in *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, 2007, no. October 2007, pp. 522–527.

[213] P. Hoek, “Parallel arc diagrams: Visualizing temporal interactions,” *J. Soc. Struct.*, vol. 12, no. 7, 2011.

[214] N. Henry, J.-D. Fekete, and M. J. McGuffin, “NodeTrix: a hybrid visualization of social networks.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 13, no. 6, pp. 1302–9, 2007.

[215] M. Ghoniem, J.-D. Fekete, and P. Castagliola, “On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis,” *Inf. Vis.*, vol. 4, no. 2, pp. 114–135, May 2005.

[216] N. Henry and J. Fekete, “MatLink : Enhanced Matrix Visualization for Analyzing Social Networks,” *Lect. Notes Comput. Sci. (Proceedings 13th IFIP TC13 Int. Conf. Human-Computer Interact. INTERACT’07)*, no. 4663, pp. 288–302, 2007.

[217] S. Zhao, M. J. McGuffin, and M. H. Chignell, “Elastic hierarchies: combining treemaps and node-link diagrams,” in *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, 2005, pp. 57–64.

[218] Q. V. Nguyen and M. L. Huang, “EncCon: an approach to constructing interactive visualization of large hierarchical data,” *Inf. Vis.*, vol. 4, no. 1, pp. 1–21, Mar. 2005.

[219] D. Holten, “Hierarchical edge bundles: visualization of adjacency relations in hierarchical data.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 5, pp. 741–8, 2006.

[220] R. Keller, C. M. Eckert, and P. J. Clarkson, “Matrices or node-link diagrams: which visual representation is better for visualising connectivity models?,” *Inf. Vis.*, vol. 5, no. 1, pp. 62–76, Apr. 2006.

[221] H. Siirtola and E. Mäkinen, “Constructing and reconstructing the reorderable matrix,” *Inf. Vis.*, vol. 4, no. 1, pp. 32–48, Feb. 2005.

[222] P. C. Wong, P. Mackey, H. Foote, and R. May, “Visual Matrix Clustering of Social Networks,” *IEEE Comput. Graph. Appl.*, vol. 33, no. 4, pp. 88–96, Jul. 2013.

[223] N. H. Müller, B. Liebold, D. Pietschmann, P. Ohler, and P. Rosenthal, “Gaze into Hierarchy : A Practice-oriented Eye Tracking Study,” in *EuroRV3: EuroVis Workshop on Reproducibility, Verification, and Validation in Visualization*, 2013.

[224] F. Windhager, L. Zenk, and P. Federico, “Visual Enterprise Network Analytics - Visualizing Organizational Change,” *Procedia - Soc. Behav. Sci.*, vol. 22, no. 0, pp. 59–68, Jan. 2011.

[225] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, “Effectiveness of animation in trend visualization.,” *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 6, pp. 1325–32, 2008.

[226] M. Smuc, P. Federico, F. Windhager, W. Aigner, L. Zenk, and S. Miksch, “How

do you connect moving dots? Insights from user studies on dynamic network visualizations," in *Handbook of Human Centric Visualization*, W. Huang, Ed. New York: Springer Science+Business Media, 2014, pp. 623–650.

[227] F. van Ham and A. Perer, "" Search, Show Context, Expand on Demand: Supporting Large Graph Exploration with Degree-of-Interest," *Vis. Comput. Graph. IEEE Trans.*, vol. 15, no. 6, pp. 953–960, 2009.

[228] J. Ahn, M. Taieb-Maimon, A. Sopan, C. Plaisant, and B. Shneiderman, "Temporal visualization of social network dynamics: Prototypes for nation of neighbors," *Soc. Comput. Behav. Model. Predict.*, pp. 309–316, 2011.

[229] M. Graham and J. Kennedy, "Multiform Views of Multiple Trees," *2008 12th Int. Conf. Inf. Vis.*, pp. 252–257, Jul. 2008.

[230] C. Ahlberg and B. Shneiderman, "Visual information seeking: tight coupling of dynamic query filters with starfield displays," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems (CHI '94)*, 1994, pp. 313–317.

[231] B. Shneiderman and C. Dunne, "Interactive network exploration to derive insights: filtering, clustering, grouping, and simplification," in *GRAPH DRAWING Lecture Notes in Computer Science*, W. Didimo and M. Patrignani, Eds. Springer Berlin Heidelberg, 2013, pp. 2–18.

[232] C. Dunne and B. Shneiderman, "Motif Simplification : Improving Network Visualization Readability with Fan, Connector, and Clique Glyphs," in *CHI '13. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 3247–3256.

[233] M. Itoh, N. Yoshinaga, M. Toyoda, and M. Kitsuregawa, "Analysis and visualization of temporal changes in bloggers' activities and interests," in *Pacific Visualization Symposium (PacificVis)*, 2012, pp. 57–64.

[234] N. H. Riche, S. Carpendale, T. Madhyastha, N. Roussel, and T. J. Grabowski, "LinkWave: a visual adjacency list for dynamic weighted networks," in *Proceedings of the 26th Conference on l'Interaction Homme-Machine (IHM '14)*, 2014, pp. 113–122.

[235] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, "Exploratory Analysis of Time-Series with ChronoLenses.," *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2422–31, Dec. 2011.

[236] R. Bade, S. Schlechtweg, and S. Miksch, "Connecting Time-oriented Data and Information to a Coherent Interactive Visualization," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2004, pp. 105–112.

[237] T. May, M. Steiger, J. Davey, and J. Kohlhammer, "Using Signposts for Navigation in Large Graphs," vol. 31, no. 3, 2012.

[238] J. Abello, S. Hadlak, and H. Schumann, "A Modular Degree-of-Interest Specification for the Visual Analysis of Large Dynamic Networks," vol. 20, no. 3, pp. 337–350, 2014.

[239] A. Aris and B. Shneiderman, "Designing semantic substrates for visual network

exploration," *Inf. Vis.*, vol. 6, no. 4, pp. 281–300, Nov. 2007.

[240] J. Fekete and C. Plaisant, "Excentric Labeling : Dynamic Neighborhood Labeling for Data Visualization," in *CHI'99*, 1999, vol. 9, no. December 1998, pp. 512–519.

[241] L. Leydesdorff and T. Schank, "Dynamic animations of journal maps: Indicators of structural changes and interdisciplinary developments," *J. Am. Soc. Inf. Sci. Technol.*, vol. 59, no. 11, pp. 1810–1818, Sep. 2008.

[242] P. Eades, W. Lai, K. Misue, and K. Sugiyama, *Preserving the mental map of a diagram*. International Institute for Advanced Study of Social Information Science, Fujitsu Limited, 1991.

[243] C. Friedrich and P. Eades, "The Marey Graph Animation Tool Demo," *Graph Draw. Lect. Notes Comput. Sci.*, vol. 1984, pp. 396–406, 2001.

[244] P. Federico, W. Aigner, S. Miksch, J. Pfeffer, M. Smuc, F. Windhager, and L. Zenk, "ViENA : Visual Enterprise Network Analytics," p. 2012, 2012.

[245] M. Freire, C. Plaisant, B. Shneiderman, and J. Golbeck, "ManyNets : An Interface for Multiple Network Analysis and Visualization," in *CHI 2010*, 2010, pp. 213–222.

[246] B. Alper, B. Bach, N. H. Riche, T. Isenberg, and J.-D. Fekete, "Weighted graph comparison techniques for brain connectivity analysis," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 483–492.

[247] J. Heer and D. boyd, "Vizster: visualizing online social networks," *IEEE Symp. Inf. Vis. 2005. INFOVIS 2005.*, pp. 32–39, 2005.

[248] B. Lee, M. Czerwinski, G. Robertson, and B. B. Bederson, "Understanding research trends in conferences using paperLens," *CHI '05 Ext. Abstr. Hum. factors Comput. Syst. - CHI '05*, pp. 1969–1972, 2005.

[249] B. Staats, G. M. Namata Jr, L. Getoor, and B. Shneiderman, "A Dual-View Approach to Interactive Network Visualization," in *CIKM '07 Proceedings of the sixteenth ACM conference on information and knowledge management*, 2007, pp. 939–942.

[250] T. Von Landesberger, M. Görner, R. Rehner, and T. Schreck, "A System for Interactive Visual Analysis of Large Graphs Using Motifs in Graph Editing and Aggregation," *Vision, Model. Vis.*, pp. 331–340, 2009.

[251] C. Ware and R. Bobrow, "Supporting visual queries on medium-sized node–link diagrams," *Inf. Vis.*, vol. 4, no. 1, pp. 49–58, Mar. 2005.

[252] T. Munzner and G. Robertson, "Constellation: A visualization tool for linguistic queries from MindNet," in *IEEE Symposium on Information Visualization*, 1999, pp. 132–135.

[253] F. Van Ham, M. Wattenberg, and F. B. Viégas, "Mapping text with phrase nets," *IEEE Trans. Vis. Comput. Graph.*, vol. 15, no. 6, pp. 1169–1176, 2009.

[254] N. Kerracher, J. Kennedy, and K. Chalmers, "The Design Space of Temporal Graph Visualisation," in *Proceedings of the Eurographics Conference on*

Visualization (EuroVis '14), Short Papers Track, 2014, pp. 7–11.

[255] P. Saraiya, C. North, V. Lam, and K. a Duca, "An insight-based longitudinal study of visual analytics.," *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 6, pp. 1511–22, 2006.

[256] M. Q. Wang Baldonado, A. Woodruff, and A. Kuchinsky, "Guidelines for using multiple views in information visualization," *Proc. Work. Conf. Adv. Vis. interfaces - AVI '00*, pp. 110–119, 2000.

[257] J. C. Roberts, "State of the Art: Coordinated & Multiple Views in Exploratory Visualization," in *Fifth International Conference on Coordinated and Multiple Views in Exploratory Visualization (CMV 2007)*, 2007, pp. 61–71.

[258] B. Meiguins and A. Meiguins, "Multiple coordinated views supporting visual analytics," in *Proceedings of the ACM SIGKDD Workshop on Visual Analytics and Knowledge Discovery: Integrating Automated Analysis with Interactive Exploration (VAKD '09).*, 2009, pp. 40–45.

[259] J. C. Roberts, "On encouraging multiple views for visualization," *Proceedings. 1998 IEEE Conf. Inf. Vis. An Int. Conf. Comput. Vis. Graph. (Cat. No.98TB100246)*, pp. 8–14, 1998.

[260] M. Wybrow, N. Elmqvist, J.-D. Fekete, T. Von Landesberger, J. J. van Wijk, and B. Zimmer, "Interaction in the Visualization of Multivariate Networks," in *Multivariate Network Visualization, LNCS 8380*, 2014, pp. 97–125.

[261] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. D. Fekete, "GraphDice: A system for exploring multivariate social networks," *Eurovis Eurographics IEEE Symp. Vis.*, vol. 29, no. 3, pp. 863–872, 2010.

[262] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury, "Capturing and supporting the analysis process," in *Visual Analytics Science and Technology, 2009. VAST 2009. IEEE Symposium on*, 2009, pp. 131–138.

[263] N. Andrienko, "Personal communication with Natalia Andrienko, 28/11/2012 and 02/01/2013." .

Appendix A **Formal Task Notation**

Tasks in the Andrienko framework are specified using a formal notation. Known items (constraints) are indicated in bold, while unknown items (targets) are indicated in italics.

Descriptive Tasks

An **elementary direct lookup task** is written:

$?y$: $\boldsymbol{f}(\mathbf{r}) = y$

Where:

- $?y$ indicates the task target (in this case, the unknown characteristic value, y)
- $\boldsymbol{f}$ is the data function
- $\mathbf{r}$ is the specified referential component.

For an **inverse lookup task**, the formalism is:

$?x$: $\boldsymbol{f}(x) = \mathbf{c}$

Where:

- $?x$ indicates the target (the unknown referrer, x)
- $\mathbf{c}$ is a specified element of the characteristic set

For synoptic tasks, the general formal description for **behaviour characterisation** is given:

$?p$: $\boldsymbol{\beta}(\boldsymbol{f}(x) \mid x \in \mathbf{R}) \approx p$

Where:

- $\boldsymbol{\beta}(\boldsymbol{f}(x) \mid x \in \mathbf{R})$ denotes the behaviour of a data function $\boldsymbol{f}$ over a reference set $\mathbf{R}$
- p is a variable representing an unknown pattern that we wish to find
- ≈ is used to indicate that the pattern "approximates" the behaviour

Pattern search is described:

$?R$: $\boldsymbol{\beta}(\boldsymbol{f}(x) \mid x \in R) \approx \mathbf{P}$

Where:

- $\mathbf{P}$ is a specified pattern
- $\boldsymbol{\beta}(\boldsymbol{f}(x) \mid x \in R)$ denotes the behaviour of a data function $\boldsymbol{f}$ over a reference set R
- R is the reference (sub)set we wish to find

The general formal description for an **elementary direct comparison task**, where we want to find two characteristics (associated with the same attribute) corresponding to two different specified references, and compare them, is given as:

$?y_1, y_2, \lambda$: $\boldsymbol{f}(\mathbf{r}_1) = y_1$; $\boldsymbol{f}(\mathbf{r}_2) = y_2$; $y_1 \; \lambda \; y_2$

Where

- $\boldsymbol{f}(\mathbf{r}_1) = y_1$ is a direct lookup task to find an unknown characteristic (y_1) corresponding to the specified reference ($\mathbf{r}_1$)
- $\boldsymbol{f}(\mathbf{r}_2) = y_2$ is a direct lookup task to find an unknown characteristic (y_2)corresponding to the specified reference ($\mathbf{r}_2$)
- λ is the unknown relation between y_1 and y_2

For **inverse comparison***,* the general formal description is:

$?x_1, x_2, \lambda: \boldsymbol{f}(x_1) \in \mathbf{C'}; \boldsymbol{f}(x_2) \in \mathbf{C''}; x_1\ \lambda\ x_2$

Where

- $\boldsymbol{f}(x_1) \in \mathbf{C'}$ and $\boldsymbol{f}(x_2) \in \mathbf{C''}$ are the inverse lookup tasks that need to be performed to find the references *x1* and *x2,* before the comparison can take place
- **C′** and **C″** are subsets of specified characteristics (note that these subsets may correspond to single values)
- λ is the unknown relation between x_1 and x_2

For synoptic tasks, a **direct behaviour comparison** task is written:

$?p1, p2, \lambda: \boldsymbol{\beta_1} \approx p1; \boldsymbol{\beta_2} \approx p_2; p_1\ \lambda\ p_2$

Where

- $\boldsymbol{\beta_1}$and $\boldsymbol{\beta_2}$ are two behaviours
- p_1 and p_2 are patterns approximating behaviours
- λ is the relation between the patterns (and therefore the behaviours) to be determined

For **synoptic inverse comparison**:

$?R_1, R_2, \lambda: \boldsymbol{\beta}(\boldsymbol{f_1}(x) \mid x \in R_1) \approx \mathbf{P}_1; \boldsymbol{\beta}(\boldsymbol{f_2}(x) \mid x \in R_2) \approx \mathbf{P}_2; R_1\ \lambda\ R_2$

Where:

- $\boldsymbol{\beta}(\boldsymbol{f_1}(x) \mid x \in R_1) \approx \mathbf{P}_1$ and $\mathbf{P}_1; \boldsymbol{\beta}(\boldsymbol{f_2}(x) \mid x \in R_2) \approx \mathbf{P}_2$ are pattern search tasks to find reference sets R_1 and R_2 ($\boldsymbol{f_1}(x)$ and $\boldsymbol{f_2}(x)$ could be two different attributes or the same attribute)
- λ is the relation between the reference sets which we want to find

The general formal definition of **relation-seeking** is given:

$?y_1, y_2, x_1, x_2: \boldsymbol{f}(x_1) = y_1; \boldsymbol{f}(x_2) = y_2; y_1 \boldsymbol{\Lambda} y_2$

Where:

- $\boldsymbol{f}(x_1) = y_1$ and $\boldsymbol{f}(x_2) = y_2$ are direct lookup tasks
- **Λ** is the specified relation between y_1 and y_2

The equivalent synoptic task can be written:

$$?R_1, R_2, p_1, p_2: \boldsymbol{f}(x) \mid x \in R_1) \approx p_1; (\boldsymbol{f}(x) \mid x \in R_2) \approx p_2; p_1 \boldsymbol{\Lambda}\, p_2$$

Where:

- R_1 and R_2 stand for the unknown reference subsets
- p_1 and p_2 are the behaviours of the attribute $\boldsymbol{f}(x)$ based on these two subsets
- **Λ** is the specified relation that must exist between p_1 and p_2

Connection Discovery

Heterogeneous behaviours

The formal notation to describe the behaviour involving two (or more) different attributes defined on the *same* reference set is:

$$\rho(\boldsymbol{f}_1(x), \boldsymbol{f}_2(x) \mid x \in \mathbf{R})$$

Where $\boldsymbol{f}_1(x)$ and $\boldsymbol{f}_2(x)$ are two attributes defined on the same reference set **R**.

Where *different* reference sets are involved, the notation is:

$$\rho(\boldsymbol{f}_1(x), \boldsymbol{f}_2(z) \mid x \in \mathbf{R}, z \in \mathbf{Z})$$

Where $\boldsymbol{f}_1(x)$ is an attribute defined on reference set **R**, and $\boldsymbol{f}_2(z)$ is an attribute defined on a different reference set, **Z**.

Homogenous behaviours

The formal notation given to describe homogenous behaviours, which involve internal connections within a single phenomenon, is:

$$\rho(\boldsymbol{f}(x), \boldsymbol{f}(x') \mid x \in \mathbf{R}_1, x' \in \mathbf{R}_2)$$

Where $\mathbf{R}_1$ and $\mathbf{R}_2$ are subsets of reference set $\mathbf{R}$.

Appendix B Task Design Space

In order to capture the variations in tasks in the temporal graph case, a set of task matrices were constructed, one for each of the main task types (lookup, comparison, relation seeking). The comparison and relation seeking matrices can also be found in their complete form at http://www.iidi.napier.ac.uk/c/downloads/downloadid/13377254 for easier reading and printing.

Formal Notation

This section provides a brief summary of the formal notation used to represent variations in tasks in the framework when applied to temporal graphs.

Data function applied to temporal graphs

In the case of temporal graphs, the following formalism is used to represent the Andrienko data function which maps a graph element at a particular time point to the corresponding values of the attributes in the data set:

$$\boldsymbol{f}(t, g) = (y_1, y_2, \ldots, y_N)$$

Where:

t represents a time point

g represents a graph element (node, edge, graph object)

$y_1, y_2, \ldots, y_N$ represents the N attributes in the data set

Key to formal notation

Bold	a specified value (constant)
Italics	an unknown value (variable)
t	a time point
T′	a (sub)set of time points/a time interval
g	a graph element (node, edge, graph object)
G′, G″	a (sub)set of graph elements
y	the value of an unknown characteristic
c	a specified characteristic
C′	a subset of characteristics
Λ, Ψ, Φ, λ, ψ, ϕ	a relation (e.g. $y_1\ \lambda\ y_2$ can be read as 'the relation between' y_1 and y_2)
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 \in \mathbf{T'})$	the behaviour $\boldsymbol{\beta}$ of a data function $\boldsymbol{f}$ over the set of graph objects **G′**, and time interval **T′**, where x_1 is a graph object in the set of graph objects (**G′**) and x_2 is a time point in the time interval (**T′**)
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T})] \mid x_1 \in \mathbf{G}\}$ $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G})] \mid x_2 \in \mathbf{T}\}$	formulae representing the two aspectual behaviours: the behaviour of the temporal behaviours (trends) over the graph (i.e. the distribution of temporal behaviours over the graph), and the behaviour over time of the behaviours (distributions) of attribute values over the set of graph elements (i.e. the temporal trend in the distribution of the attribute values).
P	a known pattern
p	An unknown pattern
≈	'approximates'

Lookup

A quadrant-level overview of the lookup task matrix is given in . The task matrix is given in .

	Graph Elements (nodes, edges, graph objects)	Graph subsets
Time Points	**Q1 Elementary** **Task components:** Referrers are graph elements and time points; characteristics are attribute values. **Direct lookup** $?y: \boldsymbol{f}(\mathbf{t}, \mathbf{g}) = y$ Involves finding the attribute value of a given graph element at a given time point. **Inverse lookup** $?\ t, g: \boldsymbol{f}(t, g) = \mathbf{c}$ Involves finding the graph element(s)/time point(s) associated with a given attribute value	**Q2 Synoptic** **Task components:** The referential component involves the whole graph (or a subset of the graph) and a single time point; behaviour is that of an attribute over the graph (at a single time). **Behaviour characterisation** $?p: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G}, x_2 = \mathbf{t}) \approx p$ Involves finding the pattern which approximates the behaviour of an attribute over the graph (or a specified subset of the graph) at the given time point **Pattern search** $?G, t: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G, x_2 = t) \approx \mathbf{P}$ Involves finding the time point(s) and/or subset(s) of graph elements over which a given pattern of attributes occur.

	Graph Elements (nodes, edges, graph objects)	**Graph subsets**
Time Intervals	**Q3 Synoptic** **Task components:** The referential component involves the whole time period (or a time interval) and a single graph element; behaviour is that of an attribute of a single graph element over time. **Behaviour characterisation** $?p$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T}) \approx p$ Involves finding the pattern which approximates the behaviour of an attribute of a given graph element over the whole time period (or a specified time interval) **Pattern search** $?g, T$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g, x_2 \in T) \approx \mathbf{P}$ Involves finding the graph element(s) and/or time interval(s) over which a given pattern of attributes occurs.	**Q4 Synoptic** **Task components:** The referential component involves the whole time period (or a time interval) and the whole graph (or a subset of the graph); behaviour is either of the two aspectual behaviours: the distribution of temporal trends over the graph or the distributions of an attribute over the graph, over time. **Behaviour characterisation** Involves finding the pattern that approximates the aspectual behaviours: $?p$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T})] \mid x_1 \in \mathbf{G}\} \approx p$ the behaviour of the temporal behaviours (trends) over the graph (i.e. the distribution of temporal behaviours over the graph) or $?p$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G})] \mid x_2 \in \mathbf{T}\} \approx p$ the behaviour over time of the behaviours (distributions) of attribute values over the set of graph objects (i.e. the temporal trend in the distribution of the attribute values); in both cases we may be interested in the behaviour associated with a given subset of the time period or the graph. **Pattern search** $?T, G$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T)] \mid x_1 \in G\} \approx \mathbf{P}$ or $?G, T$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G)] \mid x_2 \in T\} \approx \mathbf{P}$ Involves finding the subset(s) of time and/or graph elements over which a (sub)pattern of an aspectual behaviour occurs.

Figure 98 Quadrant-level overview of the lookup task matrix

		Graph Elements		Graph subsets	
		Constraint	**Target**	**Constraint**	**Target**
Time point	**Target**	**Direct look up** given a graph object and time, find the attribute value $?y: \boldsymbol{f}(\mathbf{t}, \mathbf{g}) = y$	**Inverse lookup** given an attribute value and a time point, find the graph object(s) which have this value $?\,g: \boldsymbol{f}(\mathbf{t}, g) = \mathbf{c}$	**Behaviour characterisation** Find the pattern that approximates (i.e. characterise) the behaviour of an attribute over the graph (or a subset of the graph) at the given time point $?p: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G}, x_2 = \mathbf{t}) \approx p$	**Pattern search** find the subset(s) of the graph over which a particular pattern of attribute values occurs, at the given time point $?G: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G, x_2 = \mathbf{t}) \approx \mathbf{P}$
	Constraint	**Inverse look up** given a graph object and attribute value, find the time point(s) at which it occurs $?\,t: \boldsymbol{f}(t, \mathbf{g}) = \mathbf{c}$	**Inverse lookup** given an attribute value, find the graph object(s), and the time point(s), at which the value occurs $?\,t, g: \boldsymbol{f}(t, g) = \mathbf{c}$	**Pattern search** find the time point(s) at which a particular pattern of attributes over the graph occurs $?\,t: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G}, x_2 = t) \approx \mathbf{P}$	**Pattern search** find the time point(s) and subset(s) of the graph over which a particular pattern of attribute values occurs $?G, t: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G, x_2 = t) \approx \mathbf{P}$ e.g. find (connected) subsets of the graph which have very similar attribute values, and the time points at which they occur
Time interval	**Target**	**Behaviour characterisation** characterise the behaviour of a attribute of a single node over time. $?p: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T}) \approx p$	**Pattern search** find the node(s) over which a particular pattern of attribute values occurs, over the given time interval. $?g: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g, x_2 \in \mathbf{T}) \approx \mathbf{P}$	**Behaviour characterisation** (i) characterise the behaviour of the temporal trends over the graph (i.e. the distribution of temporal behaviours over the graph) $?p: \boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T})] \mid x_1 \in \mathbf{G}\} \approx p$ (ii) characterise the behaviour of the attribute values over the graph, over time $?p: \boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G})] \mid x_2 \in \mathbf{T}\} \approx p$	**Pattern search** (i)Find the subset(s) of graph elements over which a given pattern in the collection of temporal trends occurs, over the given time interval $?\,G: \boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T})] \mid x_1 \in G\} \approx \mathbf{P}$ (ii) find the subset(s) of the graph over which a given (temporal) pattern in the pattern of attribute values over the graph occurs $?G: \boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G)] \mid x_2 \in \mathbf{T}\} \approx \mathbf{P}$
	Constraint	**Pattern search** find the time interval over which a given pattern of attribute values occurs for a given node. $?T: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in T) \approx \mathbf{P}$	**Pattern search** find the node(s) and time interval(s) over which the specified pattern of attribute values occurs $?g, T: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g, x_2 \in T) \approx \mathbf{P}$	**Pattern search** (i)Find the time interval(s) over which a given pattern in the collection of temporal trends occurs $?\,T: \boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T)] \mid x_1 \in \mathbf{G}\} \approx \mathbf{P}$ (ii) find the time interval(s) over which a given (temporal) pattern in the pattern of attribute values over the graph occurs $?T: \boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G})] \mid x_2 \in T\} \approx \mathbf{P}$	**Pattern search** (i) Find the subset(s) of graph elements and time interval(s) over which a given pattern in the collection of temporal trends occurs $?T, G: \boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T)] \mid x_1 \in G\} \approx \mathbf{P}$ (ii) Find the time interval(s) and subset(s) of the graph over which a given (temporal) pattern in the pattern of attribute values over the graph occurs $?G, T: \boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G)] \mid x_2 \in T\} \approx p$

Figure 99 Lookup task matrix

Comparison

A quadrant-level overview of the comparison task matrix is given in . Due to issues of space on the printed page, each quadrant of the comparison task matrix is shown separately (-). The compiled task matrix can be found at http://www.iidi.napier.ac.uk/c/downloads/downloadid/13377254.

Notes on comparison task matrix:

- In the following tasks, (G', t_1) is used to specify a graph subset at a given time (as opposed to just G'). This is due to the nature of the graph referrer: as linking relations in the graph referrer may change over time, a graph object at t_1 may be quite different from "the same" graph object at t_2.

- Where both graph elements/subsets and/or both time points/intervals are unspecified, an additional constraint can be added to the task i.e. that the components in question have a specified relation between them e.g. in the case of the graph referrer, that they are the same, connected, a certain distance from one another etc. or in the case of time that they are the same, overlapping, a given distance from one another etc . Where graph elements/subsets are restricted to being the same, and the temporal component is different, these become evolutionary tasks e.g. compare the time intervals over which two patterns occur over two time intervals for the same graph object:

  $?g, T', T'', \lambda, \psi: \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g, x_2 \in T') \approx \mathbf{P_1}; \boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g, x_2 \in T'') \approx \mathbf{P_2}; T' \psi T''$
- The variations of tasks involving the same/different attributes are not shown in the task matrix, but all tasks (with the exception of direct comparisons involving the same time point/interval and graph element/subset) could potentially be formulated to consider comparison involving the same attributes or two different attributes in the lookup subtask.

	Graph Elements (nodes, edges, graph objects)	Graph subsets
Time points	**Q1 Elementary** **Direct comparison** ? y_1, y_2, λ: $\boldsymbol{f_1}(\mathbf{t_1}, \mathbf{g_1}) = y_1$; $\boldsymbol{f_2}(\mathbf{t_2}, \mathbf{g_2}) = y_2$; $y_1 \lambda\ y_2$ - of attribute values associated with a given graph element at a given time (the attribute involved in the lookup tasks may be the same or different, hence the data functions $\boldsymbol{f_1}(\mathbf{x})$ and $\boldsymbol{f_2}(\mathbf{x})$). *Relations:* • *between attribute values are domain dependent.* **Inverse comparison** ? $t_1, t_2, g_1, g_2, \lambda$: $\boldsymbol{f}(t_1, g_1) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, g_2) \in \mathbf{C''}$; $(t_1, g_1)\ \lambda(t_2, g_2)$ - of two graph elements and/or two time points associated with given attribute values *Relations:* • *between graph elements: equality (same/different element); set relations (between the sets of elements belonging to graph objects); equality of configuration (in graph objects); linking (between nodes/graph objects, at a single time point only);* • between *two time points: happens before(/after), happens at the same time* [49].	**Q2 Synoptic** **Direct comparison** ? p_1, p_2, λ: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''}, x_2 = \mathbf{t_2}) \approx p_2$; $p_1 \lambda\ p_2$ **–** of two patterns of an attribute(s)[37] over the graph (or a subset of the graph elements) at given time point(s) *Relations:* • *between patterns: same(similar)/different/opposite*[38] **Inverse comparison** ? $G', G'', t_1, t_2, \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = t_1) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx \mathbf{P_2}$; $(G', t_1)\ \lambda\ (G'', t_2)$; $t_1 \psi\ t_2$ **-** of the time points at which the given patterns occur - of the graph subsets over which a given pattern occurs; - comparison of both time points and graph subsets. *Relations:* • *between two time points: happens before(/after), happens at the same time* [49]*;* • *between two graph subsets: equality (same/different subset); set relations (between the sets of nodes/edges belonging to the subset); equality of configuration (of the subset); linking (between nodes/graph objects, at a single time point only).*
Time intervals	**Q3 Synoptic** **Direct comparison** ? p_1, p_2, λ: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_2}, x_2 \in \mathbf{T''}) \approx p_2$; $p_1 \lambda\ p_2$ - of two (temporal) patterns associated with an attribute(s)[Error! Bookmark not defined.] of given graph element(s) over the whole time period (or a specified time interval) *Relations:* • *between patterns: same (similar)/different/opposite*	**Q4 Synoptic** **Direct comparison** ? p_1, p_2, λ: $\boldsymbol{\beta_G}\{\boldsymbol{\beta_T}[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_1$; $\boldsymbol{\beta_G}\{\boldsymbol{\beta_T}[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in \mathbf{G''}\} \approx p_2$; $p_1 \lambda\ p_2$ (comparison of patterns of distributions of temporal trends over the graph) or ? p_1, p_2, λ: $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_1$; $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in \mathbf{T''}\} \approx p_2$; $p_1 \lambda\ p_2$ (comparison of patterns of distributions of an attribute over the graph, over time)

[37] i.e. each pattern may correspond to a different attribute

[38] In descriptive synoptic tasks (in connectional synoptic tasks, patterns of "mutual" behaviours include correlation, dependency, and structural connection.

	Graph Elements (nodes, edges, graph objects)	Graph subsets
	Inverse comparison ? g_1, g_2, T', T'', λ, ψ: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P_2}$; $g_1 \lambda g_2$; $T' \psi T''$ – of the time intervals over which given patterns occur; of the graph elements associated with a given pattern; comparison of both time intervals and graph elements *Relations:* • *between two graph elements: equality (same/different; set relations between the sets of elements belonging to graph objects);* • *between time intervals: happens before(/after), happens at the same time; between two intervals, or an instant and an interval: happens before(/after), starts, finishes, happens during; between intervals only: overlaps, meets* [49].	– of two patterns associated with a given subset of time and/or subset of graph elements. The patterns may reflect either of the two aspectual behaviours (the distribution of temporal trends over the graph or the distributions of an attribute over the graph, over time) *Relations* • *between patterns: same (similar)/different/opposite* **Inverse comparison** ? G', G'', T', T'', λ, ψ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in G'\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $T' \lambda T''$; $G' \psi G''$; or ? G', G'', T', T'', λ, ψ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in T'\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P_2}$; $T' \lambda T''$; $G' \psi G''$; – of the time intervals and/or subsets of graph elements associated with a given aspectual (sub)pattern *Relations:* • *between two graph subsets: equality, set relations* • *between time intervals: happens before(/after), happens at the same time; between two intervals, or an instant and an interval: happens before(/after), starts, finishes, happens during; between intervals only: overlaps, meets* [49].

Figure 100 Quadrant-level overview of the comparison task matrix

			Graph elements (nodes, edges, graph objects)			
			Both constraints		One element specified	Neither element specified
			Single/same element	Two different elements		
Time points	Both constraints	Same time	**Direct comparison** Compare the values of *different* attributes for a given node at a given time point. ? y_1, y_2, λ: $\boldsymbol{f_1}(\mathbf{t}, \mathbf{g}) = y_1$; $\boldsymbol{f_2}(\mathbf{t}, \mathbf{g}) = y_2$; $y_1 \lambda y_2$	**Direct comparison** Compare the attribute values associated with two different nodes at the same time point. ? y_1, y_2, λ: $\boldsymbol{f}(\mathbf{t}, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(\mathbf{t}, \mathbf{g_2}) = y_2$; $y_1 \lambda y_2$	**Inverse comparison** This task reduces to comparison with a specified reference[i] . Find and compare with a given node, the node(s) associated with the given attribute value at the given time. ? g_2, λ: $\boldsymbol{f}(\mathbf{t}, g_2) \in \mathbf{C'}$; $(\mathbf{t}, \mathbf{g_1}) \lambda (\mathbf{t}, g_2)$	**Inverse comparison** Find and compare the nodes associated with two different attribute values at the given time ? g_1, g_2, λ: $\boldsymbol{f}(\mathbf{t}, g_1) \in \mathbf{C'}$; $\boldsymbol{f}(\mathbf{t}, g_2) \in \mathbf{C''}$; $(\mathbf{t}, g_1) \lambda (\mathbf{t}, g_2)$
		Different times	**Direct comparison** Compare the attribute values associated with a single node at two different times. ? y_1, y_2, λ: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(\mathbf{t_2}, \mathbf{g_2}) = y_2$; $y_1 \lambda y_2$	**Direct comparison** Compare the attribute values associated with two different nodes at two different times. ? y_1, y_2, λ: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(\mathbf{t_2}, \mathbf{g_2}) = y_2$; $y_1 \lambda y_2$	**Inverse comparison** As above but involving two different time points[ii]. Find and compare with a given node, the node(s) associated with the given attribute value at the given times. ? g_2, λ: $\boldsymbol{f}(\mathbf{t_2}, g_2) \in \mathbf{C'}$; $(\mathbf{t_1}, \mathbf{g_1}) \lambda (\mathbf{t_2}, g_2)$	**Inverse comparison** As above, but involving two different time points. Find and compare the nodes associated with two different attribute values at the given times ? g_1, g_2, λ: $\boldsymbol{f}(\mathbf{t_1}, g_1) \in \mathbf{C'}$; $\boldsymbol{f}(\mathbf{t_2}, g_2) \in \mathbf{C''}$; $(\mathbf{t_1}, g_1) \lambda (\mathbf{t_2}, g_2)$

	Graph elements (nodes, edges, graph objects)			
	Both constraints		One element specified	Neither element specified
	Single/same element	Two different elements		
One time point specified	**Inverse comparison** This task reduces to comparison with a specified reference[iii]. Find the time point(s) associated with the given attribute value for the given node, and compare it with a given time point. ? t_2, λ: $\boldsymbol{f}(t_2, \mathbf{g}) \in \mathbf{C'}$; $\mathbf{t_1}\ \lambda\ t_2$	**Inverse comparison** As left, this task reduces to comparison with a specified reference[iv]. ? t_2, λ: $\boldsymbol{f}(t_2, \mathbf{g}) \in \mathbf{C'}$; $\mathbf{t_1}\ \lambda\ t_2$	**Inverse comparison** Either: A task reduced to comparison with a specified reference[v]. Find the node(s) and time point(s) at which it has a given attribute value, and compare this with a given node at a given time point. ? t_2, g_2, λ, Ψ: $\boldsymbol{f}(t_2, g_2) \in \mathbf{C'}$; $(\mathbf{t_1}, \mathbf{g_1})\ \lambda (t_2, g_2)$; $\mathbf{t_1}\ \Psi\ t_2$ OR Find the time point at which a given node has a given attribute value, and the node which has a given attribute value at a given time, and compare the nodes and time points. ? t_1, g_2, λ, Ψ: $\boldsymbol{f}(t_1, \boldsymbol{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(\mathbf{t_2}, g_2) \in \mathbf{C''}$; $(t_1, \boldsymbol{g_1})\ \lambda (\mathbf{t_2}, g_2)$; $t_1\ \Psi\ \mathbf{t_2}$	**Inverse comparison** Find the node(s) having a specified attribute value at a given time, and the node(s) and time point(s) having a given attribute value, and compare the nodes and time points. ? t_2, g_1, g_2, λ: $\boldsymbol{f}(\mathbf{t_1}, g_1) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, g_2) \in \mathbf{C''}$; $(\mathbf{t_1}, g_1)\ \lambda (t_2, g_2)$
Neither time point specified	**Inverse comparison** Find and compare the times at which the given node had the given attribute values. ? t_1, t_2, λ: $\boldsymbol{f}(t_1, \mathbf{g}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, \mathbf{g}) \in \mathbf{C''}$; $t_1\ \lambda\ t_2$	**Inverse comparison** Find and compare the times at which two given nodes had the given attribute values. ? t_1, t_2, λ: $\boldsymbol{f}(t_1, \mathbf{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, \mathbf{g_2}) \in \mathbf{C''}$; $t_1\ \lambda\ t_2$	**Inverse comparison** Find the time point(s) at which a given node had a given attribute value, and the time point(s) and node(s) having a second given attribute value, and compare the nodes and time points. ? t_1, t_2, g_2, λ: $\boldsymbol{f}(t_1, \mathbf{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, g_2) \in \mathbf{C''}$; $(t_1, \mathbf{g_1})\ \lambda (t_2, g_2)$	**Inverse comparison** Find the time points and nodes associated with two given attribute values and compare them. ? $t_1, t_2, g_1, g_2, \lambda$: $\boldsymbol{f}(t_1, g_1) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, g_2) \in \mathbf{C''}$; $(t_1, g_1)\ \lambda (t_2, g_2)$

Figure 101 Comparison task matrix, quadrant 1: considers comparisons involving graph elements (nodes, edges, graph objects) and time points (i.e. the elementary comparison tasks)

			Graph subsets			
			Both constraints		One constraint, one target	Both are targets
			Same subset	Different subsets		
Time points	Both constraints	Same time	**Direct comparison** of the attribute patterns *of two different attributes* over the same subset of the graph at the same time point. ? p_1, p_2, λ: $\beta(\mathbf{f_1}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t}) \approx p_1;$ $\beta(\mathbf{f_2}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = {}_2) \approx p_2;$ $p_1 \lambda\, p_2$	**Direct comparison** of the attribute patterns over two different subsets of the graph at the same time point. ? p_1, p_2, λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t}) \approx p_1;$ $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G''}, x_2 = \mathbf{t}) \approx p_2;$ $p_1 \lambda\, p_2$	**Inverse comparison** of a given graph subset with the graph subset associated with a given pattern at a given time[vi]. ? G', λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t}) \approx \mathbf{P};$ $G', \mathbf{t})\ \lambda\ (\mathbf{G''}, \mathbf{t})$	**Inverse comparison** of two graph subsets associated with two given patterns at the same specified time. ? G', G'', λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t}) \approx \mathbf{P_1};$ $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t}) \approx \mathbf{P_2};$ $(G', \mathbf{t})\ \lambda\ (G'', \mathbf{t})$
		Different times	**Direct comparison** of the attribute patterns over the same subset of the graph at two different time points. ? p_1, p_2, λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx p_1;$ $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_2}) \approx p_2;$ $p_1 \lambda\, p_2$	**Direct comparison** of the attribute patterns over two different subsets of the graph at two different time points. ? p_1, p_2, λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx p_1;$ $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in \mathbf{G''}, x_2 = \mathbf{t_2}) \approx p_2;$ $p_1 \lambda\, p_2$	**Inverse comparison a**s above, but the specified subset of graph elements is associated with a different time point: ? G', λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t_1}) \approx \mathbf{P};$ $(G', \mathbf{t_1})\ \lambda\ (\mathbf{G''}, \mathbf{t_2});$	**Inverse comparison** of two graph subsets associated with two given patterns at two different, specified time points. ? G', G'', λ: $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t_1}) \approx \mathbf{P_1};$ $\beta(\mathbf{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t_2}) \approx \mathbf{P_2};$ $(G', \mathbf{t_1})\ \lambda\ (G'', \mathbf{t_2})$

	Graph subsets			
	Both constraints		**One constraint, one target**	**Both are targets**
	Same subset	**Different subsets**		
One constraint, one target	**Inverse comparison** of the time point associated with a given pattern over a given subset of the graph, with a given time point[39]. ? t_2, λ: $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_2) \approx \mathbf{P}$; $\mathbf{t_1}\ \lambda\ t_2$	**Inverse comparison**, as left[40]	**Inverse comparison** of a given graph subset at a given time with the graph subset associated with a given pattern, *and* comparison of a given time point with the time point also associated with the given pattern. This may involve only one lookup subtask[41] or two: ? G'', t_2, λ, ψ: $\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = t_2) \approx \mathbf{P_2}$; $(\mathbf{G'}, \mathbf{t_1})\ \lambda\ (G'', t_2)$; $\mathbf{t_1} \psi\ t_2$ or ? G'', t_1, λ, ψ: $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_1) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = \mathbf{t_2}) \approx \mathbf{P_2}$; $(\mathbf{G'}, t_1)\ \lambda\ (G'', t_2)$; $t_1 \psi\ \mathbf{t_2}$	**Inverse comparison** of the graph objects associated with two patterns, one of them occurring at a given time, *and* comparison of the given time point with the unknown time point at which the second pattern occurs. ? $G', G'', t_2, \lambda, \psi$: $\beta(f(x_1, x_2) \mid x_1 \in G',\ x_2 = \mathbf{t_1}) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = t_2) \approx \mathbf{P_2}$; $(G', \mathbf{t_1})\ \lambda\ (G'', t_2)$; $\mathbf{t_1} \psi\ t_2$
Both are targets	**Inverse comparison** of the time points at which two different patterns occur, over the same graph subset. ? t_1, t_2, λ: $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_1) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_2) \approx \mathbf{P_2}$; $t_1\ \lambda\ t_2$	**Inverse comparison** of the time points at which two different patterns occur, over two different graph subsets. ? t_1, t_2, λ: $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_1) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G''},\ x_2 = t_2) \approx \mathbf{P_2}$; $t_1\ \lambda\ t_2$	**Inverse comparison** of the graph subsets associated with given patterns, where one of the graph subsets is specified, but the time at which it occurs is unknown, the other graph subset and time at which the pattern occurs is not specified. In addition, we may wish to compare the time points at which the patterns occurred. ? $G', G'', t_1, t_2, \lambda, \psi$: $\beta(f(x_1, x_2) \mid x_1 \in \mathbf{G'},\ x_2 = t_1) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = t_2) \approx \mathbf{P_2}$; $(\mathbf{G'}, t_1)\ \lambda\ (G'', t_2)$; $t_1 \psi\ t_2$	**Inverse comparison** of the graph subsets and time points associated with two given patterns. ? $G', G'', t_1, t_2, \lambda, \psi$: $\beta(f(x_1, x_2) \mid x_1 \in G',\ x_2 = t_1) \approx \mathbf{P_1}$; $\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = t_2) \approx \mathbf{P_2}$; $(G', t_1)\ \lambda\ (G'', t_2)$; $t_1 \psi\ t_2$

Figure 102 Comparison quadrant 2: considers comparisons involving the behaviour of an attribute over the graph (or a graph subset)

			Graph elements (nodes, edges, graph objects)			
			Both graph elements specified		One graph element specified	Neither graph element specified
			Single/same graph element	Two different graph elements		
Time intervals	Both Constraints	Same interval	**Direct comparison** of the attribute patterns *of two different attributes* of the same graph element over the same time interval. $?p_1, p_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f_1}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T'}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f_2}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T'}) \approx p_2$; $p_1 \lambda\, p_2$	**Direct comparison** of the patterns of two different graph elements over the same time interval $?p_1, p_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_2}, x_2 \in \mathbf{T'}) \approx p_2$; $p_1 \lambda\, p_2$	**Inverse comparison** of a graph element associated with a given pattern over a given time interval, with a given graph element.[42] $?g_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx \mathbf{P}$; $\mathbf{g_1} \lambda\, g_2$	**Inverse comparison** of two graph elements associated with given patterns over the same given time interval. $?g_1, g_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx \mathbf{P_2}$; $g_1 \lambda\, g_2$
		Different intervals	**Direct comparison** of the patterns of the same graph element over two different time intervals. $?p_1, p_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T'}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T''}) \approx p_2$; $p_1 \lambda\, p_2$	**Direct comparison** of the patterns of two different graph elements over two different time intervals. $?p_1, p_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx p_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_2}, x_2 \in \mathbf{T''}) \approx p_2$; $p_1 \lambda\, p_2$	**Inverse comparison** as above[43].	**Inverse comparison** of two graph elements associated with given patterns over the two different given time intervals. $?g_1, g_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T''}) \approx \mathbf{P_2}$; $g_1 \lambda\, g_2$
	One constraint, one target		**Inverse comparison** of a time interval associated over which a given pattern occurs for a given graph element, with a specified time interval.[44] $?T'', \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in T'') \approx \mathbf{P}$; $\mathbf{T'} \lambda\, T''$	**Inverse comparison** as left[45].	**Inverse comparison** of a given graph element with a graph element associated with a given pattern (over a time interval which may or may not be specified) *and* comparison of a given time interval with a time interval associated with a given pattern (which may or may not be associated with a given graph element). This may involve only one lookup subtask[46] or two:	**Inverse comparison** of two graph elements associated with given patterns (one of which is a pattern over a specified time interval) *and* comparison of the time intervals over which the patterns occur. $?g_1, g_2, T'', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P_2}$; $g_1 \lambda\, g_2$; $\mathbf{T'} \psi\, T''$

[42] Reduced from: $?g_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx \mathbf{P_2}$; $\mathbf{g_1} \lambda\, g_2$

[43] Reduced from $?g_2, \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T''}) \approx \mathbf{P_2}$; $\mathbf{g_1} \lambda\, g_2$

[44] Reduced from: $?T'', \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in T'') \approx \mathbf{P_2}$; $\mathbf{T'} \lambda\, T''$

[45] Reduced from: $?T'', \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_2}, x_2 \in T'') \approx \mathbf{P_2}$; $\mathbf{T'} \lambda\, T''$

[46] Reduced from: $?g_2, T', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P_2}$; $\mathbf{g_1} \lambda\, g_2$; $\mathbf{T'} \psi\, T''$

	Graph elements (nodes, edges, graph objects)			
	Both graph elements specified		One graph element specified	Neither graph element specified
	Single/same graph element	Two different graph elements		
			$?g_2, T'', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P}$; $\mathbf{g_1}\, \lambda\, g_2$; $\mathbf{T'}\, \psi\, T''$ Or $?g_2, T', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T''}) \approx \mathbf{P_2}$; $\mathbf{g_1}\, \lambda\, g_2$; $T'\, \psi\, \mathbf{T''}$	
Both are targets	**Inverse comparison** of the time intervals over which the given patterns occur for a single given graph element. $?T', T'', \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g}, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g}, x_2 \in T'') \approx \mathbf{P_2}$; $T'\, \lambda\, T''$	**Inverse comparison** of the time intervals over which the given patterns occur for two different graph elements. $?T', T'', \lambda$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g_1}, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g_2}, x_2 \in T'') \approx \mathbf{P_2}$; $T'\, \lambda\, T''$	**Inverse comparison** of a specified graph element and a graph element associated with a given pattern (over an unspecified time interval) *and* comparison of the time intervals over which the patterns occur. $?\, g_2, T', T'', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P_2}$; $\mathbf{g_1}\, \lambda\, g_2$; $T'\, \psi\, T''$	**Inverse comparison** of graph elements and time intervals associated with two given patterns. $?g_1, g_2, T', T'', \lambda, \psi$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in T') \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx \mathbf{P_2}$; $g_1\, \lambda\, g_2$; $T'\, \psi\, T''$

Figure 103 Comparison quadrant 3: considers comparisons involving the behaviour of an attribute of a single graph element over time (i.e. a temporal trend)

			Graph subsets			
			Both graph subsets specified		One graph subset specified	Neither graph subset specified
			Single/same subset	Two different subsets		
Time intervals	Both constraints	Same time	**Direct comparison** of distributions of temporal trends over the graph *for two different attributes* over the same time interval and for the same graph subset: $? p_1, p_2, \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f_1}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f_2}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_2$; $p_1 \lambda\, p_2$ Or temporal trends in distributions of an attribute over the graph *for two different attributes* for the same graph subset and over the same time interval: $? p_1, p_2, \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f_1}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f_2}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_2$; $p_1 \lambda\, p_2$	**Direct comparison** of distributions of temporal trends over two different graph subsets over the same time interval: $? p_1, p_2, \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f_1}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f_2}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G''}\} \approx p_2$; $p_1 \lambda\, p_2$ Or temporal trends in distributions of an attribute over the graph, over two different graph subsets over the same time interval: $? p_1, p_2, \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f_1}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f_2}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in \mathbf{T'}\} \approx p_2$; $p_1 \lambda\, p_2$	**Inverse comparison** of the subset of graph elements associated with a given pattern involving a given time interval, and a given subset of graph elements[47]: $? G'', \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G''\} \approx \mathbf{P}$; $\mathbf{G'}\ \lambda\ G''$; or $?G'', \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P}$; $\mathbf{G'}\ \lambda\ G''$;	**Inverse comparison** of two graph subsets associated with two given patterns involving the same time interval: $? G', G'', \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $G'\ \lambda\ G''$; or $? G', G'', \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_2}$; $G'\ \lambda\ G''$;
		Differ	**Direct comparison** of distributions of temporal trends over the graph for the	**Direct comparison** of distributions of temporal trends over two different	**Inverse comparison** as above[48]	**Inverse comparison** of two graph subsets associated with two given patterns involving two different time intervals

[47] Reduced from: $? G'', \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G''\} \approx \mathbf{P_2}$;$\mathbf{G'}\ \lambda\ G''$;
OR
$?G'', \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$;$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_2}$; $\mathbf{G'}\ \lambda\ G''$;

[48] Reduced from: $? G'', \lambda$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $\mathbf{G'}\ \lambda\ G''$;OR
$? G'', \lambda$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$;$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx \mathbf{P_2}$; $\mathbf{G'}\ \lambda\ G''$;

	Graph subsets			
	Both graph subsets specified		One graph subset specified	Neither graph subset specified
	Single/same subset	Two different subsets		
	same graph subset during two different time intervals: ? p_1, p_2, λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}_1(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}_2(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in \mathbf{G'}\} \approx p_2$; $p_1 \lambda p_2$ Or temporal trends in distributions of an attribute over the graph, for the same graph subset over two different time intervals: ? p_1, p_2, λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}_1(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}_2(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T''}\} \approx p_2$; $p_1 \lambda p_2$	graph subsets over two different time intervals: ? p_1, p_2, λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}_1(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx p_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}_2(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in \mathbf{G''}\} \approx p_2$; $p_1 \lambda p_2$ Or temporal trends in distributions of an attribute over the graph, over two different graph subsets over two different time intervals: ? p_1, p_2, λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}_1(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx p_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}_2(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in \mathbf{T''}\} \approx p_2$; $p_1 \lambda p_2$		? G', G'', λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $G' \lambda G''$; or ? G', G'', λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx \mathbf{P_2}$; $G' \lambda G''$;
One constrai	**Inverse comparison** of a time interval associated with a given pattern and graph subset, and a given time interval[49]:	**Inverse comparison** as left[50].	**Inverse comparison** of a time interval and graph subset associated with a given	**Inverse comparison** of graph subsets and time intervals associated with two given patterns, where one of the patterns involves a given time interval:

[49] Reduced from: ? T'', λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_2}$; $\mathbf{T'} \lambda T''$; or ? T'', λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T''\} \approx \mathbf{P_2}$;
$\mathbf{T'} \lambda T''$;

[50] Reduced from: ? T'', λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_1}$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx \mathbf{P_2}$;$\mathbf{T'} \lambda T''$; or ? T'', λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$;$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in T''\} \approx \mathbf{P_2}$;
$\mathbf{T'} \lambda T''$;

Graph subsets			
Both graph subsets specified		One graph subset specified	Neither graph subset specified
Single/same subset	Two different subsets		
$? T'', \lambda$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P}$; $\mathbf{T'}\ \lambda\ T''$; or $? T'', \lambda$: $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T''\} \approx \mathbf{P}$; $\mathbf{T'}\ \lambda\ T''$;		pattern, with a given time interval and graph subset[51] $? G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P}$; $\mathbf{T'}\ \lambda\ T''$; $\mathbf{G'}\ \psi\ G''$; or $? G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P}$; $\mathbf{T'}\ \lambda\ T''$; $\mathbf{G'}\ \psi\ G''$; **OR** **Inverse comparison** of a graph object associated with a pattern involving a given time interval, and a given graph object *and* a time interval associated with a pattern involving a given graph subset, and a given time interval. $? G', T', T'', \lambda, \psi$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx \mathbf{P_1}$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx \mathbf{P_2}$; $\mathbf{T'}\ \lambda\ T''$; $G'\ \psi\ \mathbf{G''}$; or $? G'', T', \lambda, \psi$:	 $? G', G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx \mathbf{P_1}$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $\mathbf{T'}\ \lambda\ T''$; $G'\ \psi\ G''$; or $? G', G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$; $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P_2}$; $\mathbf{T'}\ \lambda\ T''$; $G'\ \psi\ G''$;

[51] Reduced from: $? G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P_1}$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P_2}$; $\mathbf{T'}\ \lambda\ T''$; $\mathbf{G'}\ \psi\ G''$; or $? G'', T'', \lambda, \psi$: $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx \mathbf{P_1}$; $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P_2}$; $\mathbf{T'}\ \lambda\ T''$; $\mathbf{G'}\ \psi\ G''$;

	Graph subsets			
	Both graph subsets specified		**One graph subset specified**	**Neither graph subset specified**
	Single/same subset	**Two different subsets**		
			$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx \mathbf{P}_2$; $T' \lambda \mathbf{T''}$; $\mathbf{G'} \psi G''$;	
Both are targets	**Inverse comparison** of two time intervals associated with two given patterns involving the same graph subset: ? T', T'', λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P}_2$; $T' \lambda T''$; or ? T', T'', λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T''\} \approx \mathbf{P}_2$; $T' \lambda T''$;	**Inverse comparison** of two time intervals associated with two given patterns involving two different graph subsets: ? T', T'', λ: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx \mathbf{P}_2$; $T' \lambda T''$; or ? T', T'', λ: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in T''\} \approx \mathbf{P}_2$; $T' \lambda T''$;	**Inverse comparison** of graph subsets and time intervals associated with given patterns, where one of the patterns involves a given graph subset: ? $G'', T', T'', \lambda, \psi$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P}_2$; $T' \lambda T''$; $\mathbf{G'} \psi G''$; or ?$G'', T', T'', \lambda, \psi$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P}_2$; $T' \lambda T''$; $\mathbf{G'} \psi G''$;	**Inverse comparison** of graph subsets and time intervals associated with given patterns: ? $G', G'', T', T'', \lambda, \psi$: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in G'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx \mathbf{P}_2$; $T' \lambda T''$; $G' \psi G''$; or ? $G', G'', T', T'', \lambda, \psi$: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in T'\} \approx \mathbf{P}_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx \mathbf{P}_2$; $T' \lambda T''$; $G' \psi G''$;

Figure 104 Comparison quadrant 4: considers comparisons involving aspectual behaviours (i) the behaviour of temporal trends for all graph elements, over the graph (ii) the behaviour of an attribute over the graph, over time

Relation Seeking

A quadrant-level overview of the comparison task matrix is given in . Again, due to issues of space on the printed page, each quadrant of the relation seeking task matrix is shown separately (-). The complete task matrix can be found at http://www.iidi.napier.ac.uk/c/downloads/downloadid/13377254.

Notes on Relation Seeking matrix:

- The tasks in the matrices have been formulated to show the same attribute, but each task could also be formulated for the case where two different attributes are involved.
- Tasks where attribute values or patterns are specified are not shown in the matrix. These tasks can be formulated to produce tasks where either:

i. Both attribute values or patterns are specified. In this case, the relation seeking task will involve a specified relation on time points/intervals and/or graph elements/subsets. Taking an example from quadrant 2, we could have:

Find graph subsets and time points associated with given patterns, where the graph subsets/time points are related in the specified way.

$? \, G', G'', t_1, t_2$:
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = t_1) \approx \mathbf{P_1}$;
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx \mathbf{P_2}$;
$t_1 \, \boldsymbol{\Psi} \, t_2$;
$G' \, \boldsymbol{\Phi} \, G''$;

ii. One attribute value or pattern is specified. In this case, the specified relation may be between attribute values or patterns, graph elements or subsets and/or time points or intervals (as appropriate to the specified/unspecified elements in the task). Again, an example from quadrant 2 is given:

Find patterns related to a given pattern in the given way. Find also the graph subsets and time points over/at which the related patterns occur. A relation between graph subsets and/or time points may also be specified.

? G', G'', t_1, t_2, P_2 :
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = t_1) \approx \mathbf{P_1}$;
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$;
$t_1\ \boldsymbol{\Psi}\ t_2$;
$G'\ \boldsymbol{\Phi}\ G''$;
$P_1\ \boldsymbol{\Lambda}\ P_2$

- In the case where one of the subtasks is completely specified, the task is reduced to relation seeking involving a specified pattern or graph subset e.g.

? G'', t_2, P_2 :
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx \mathbf{P_1}$;
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$;
$\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$;
$\mathbf{G'}\ \boldsymbol{\Phi}\ G''$;
$\mathbf{P_1}\ \boldsymbol{\Lambda}\ P_2$

Can be reduced to:

Find patterns/graph elements related in the given way to given patterns/graph elements. A relation on time points may also be specified.

? G'', t_2, P_2 :
$\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$;
$\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$;
$\mathbf{G'}\ \boldsymbol{\Phi}\ G''$;
$\mathbf{P_1}\ \boldsymbol{\Lambda}\ P_2$

<table>
<tr><th></th><th>Graph elements (nodes, edges, graph objects)</th><th>Graph subsets</th></tr>
<tr><td>Time points</td><td>Elementary
? $t_1, t_2, g_1, g_2, y_1, y_2$:
$f(t_1, g_1) = y_1$;
$f(t_2, g_2) = y_2$;
$t_1 \, \Psi \, t_2$;
$g_1 \, \Phi \, g_2$;
$y_1 \wedge y_2$

Relation seeking – find the attribute values related in the given manner (and possibly the corresponding graph element(s)/time point(s)). In this case the possible relation specified is domain dependent. Variations of this task depend on the number of time points and graph elements specified in the lookup sub tasks.

Additional constraints on the relations between graph elements and/or time points may also be specified. Depending on the elements involved in the lookup tasks (i.e. whether they are specified/unspecified, same or different), constraints may be any of the relations noted in the comparison matrix e.g.:

• between time points: equality (same/different time point), that time points are consecutive, occur before/after a given time point, that a certain distance exists between them etc.
• between graph elements: equality (same/different element); set relations (between the sets of elements belonging to graph objects); equality of configuration (in graph objects); linking (where a single time point is specified in the lookup task or a constraint of equality is added on unspecified time points).</td><td>Synoptic
? $G', G'', t_1, t_2, P_1, P_2$:
$\beta(f(x_1, x_2) \mid x_1 \in G',\ x_2 = t_1) \approx P_1$;
$\beta(f(x_1, x_2) \mid x_1 \in G'',\ x_2 = t_2) \approx P_2$;
$t_1 \, \Psi \, t_2$;
$G' \, \Phi \, G''$;
$P_1 \wedge P_2$

Relation seeking – find patterns of attribute(s) over the graph which are related in the given manner (and possibly the time point(s)/subsets of graph elements at/over which they occur). Possible specified relations between patterns are same (similar)/different/opposite. Variations depend on the number of time points and graph subsets specified in the lookup subtasks.

Additional constraints on relations between time points /or graph subsets may also be included in the task specification, depending on the elements involved in the lookup tasks. These are similar to the relations noted in the comparison matrix e.g.

• between time points: equality (same/different time point), that time points are consecutive, occur before/after a given time point, that a certain distance exists between them etc.
• between two graph subsets: equality (same/different subset); set relations (between the sets of nodes/edges belonging to the subset); equality of configuration of the subset, linking (between nodes/graph objects, at a single time point only).</td></tr>
<tr><td>Time intervals</td><td>Synoptic
? $g_1, g_2; T', T'', P_1, P_2$:
$\beta(f(x_1, x_2) \mid x_1 = g_1, x_2 \in T') \approx P_1$;
$\beta(f(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx P_2$;
$g_1 \, \Phi \, g_2$; $T \, \Psi \, T''$; $P_1 \wedge P_2$

Relation seeking – find the patterns of attribute(s) over time which are related in the given manner (and possibly find the graph element(s) to which they correspond/the time period(s) over which they occur). The possible specified relations between patterns are the same (similar)/different/opposite. Variations depend on the number of graph elements and time intervals specified in the lookup subtasks.

Additional constraints on relations between graph elements and/or time intervals may also be included in the task specification, depending on the elements involved in the lookup tasks. These are similar to the relations noted in the comparison matrix e.g.</td><td>Synoptic
? $G', G'', T', T'', P_1, P_2$:
$\beta_G\{\beta_T[f(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in G'\} \approx P_1$;
$\beta_G\{\beta_T[f(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx P_2$;
$T' \, \Psi \, T''$; $G' \, \Phi \, G''$; $P_1 \wedge P_2$

or

? $G', G'', T', T'', P_1, P_2$:
$\beta_T\{\beta_G[f(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in T'\} \approx P_1$;
$\beta_T\{\beta_G[f(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx P_2$;
$T' \, \Psi \, T''$; $G' \, \Phi \, G''$; $P_1 \wedge P_2$

Relation seeking – find (sub)patterns of either of the aspectual behaviours which are related in the given manner (and possibly find the graph subset/time interval associated with the found patterns). The possible specified relations between patterns are the same (similar)/different/opposite. Variations depend on the number of graph subsets and</td></tr>
</table>

	• between graph elements: equality (same/different element); set relations (between the sets of elements belonging to graph objects); equality of configuration (in graph objects). • Between the time intervals (over which the pattern occurs): happens before(/after), happens at the same time; between two intervals, or an instant and an interval: happens before(/after), starts, finishes, happens during; between intervals only: overlaps, meets [49].	time intervals specified in the lookup subtasks. Additional constraints on relations between time points /or graph subsets may also be included in the task specification, depending on the elements involved in the lookup tasks. These are similar to the relations noted in the comparison matrix e.g. • between two graph subsets: equality (same/different subset); set relations (between the sets of nodes/edges belonging to the subset); equality of configuration of the subset, linking (between nodes/graph objects, at a single time point only). • Between the time intervals (over which the pattern occurs): happens before(/after), happens at the same time; between two intervals, or an instant and an interval: happens before(/after), starts, finishes, happens during; between intervals only: overlaps, meets [49].

Figure 105 Relation seeking quadrant-level overview

<table>
<tr><td colspan="3" rowspan="3"></td><td colspan="4">**Graph elements (nodes, edges, graph objects)**</td></tr>
<tr><td colspan="2">**Both constraints**</td><td rowspan="2">**One constraint, one target**</td><td rowspan="2">**Both are targets**</td></tr>
<tr><td>**Same element**</td><td>**Different elements**</td></tr>
<tr><td rowspan="2">**Time points**</td><td rowspan="2">**Both constraints**</td><td>**Same time**</td><td colspan="2" rowspan="2">Not applicable</td><td>Find the attribute value (and associated node) at a given time, which is related in the given way to an attribute value associated with a given graph object at the same given time point. A relation between graph elements may also be specified.

? g_2, y_1, y_2:
$\boldsymbol{f}(\mathbf{t}, \mathbf{g_1}) = y_1; \boldsymbol{f}(\mathbf{t}, g_2) = y_2;$
$y_1\, \boldsymbol{\Lambda}\, y_2$; $\boldsymbol{g_1}\, \boldsymbol{\Phi}\, g_2$</td><td>Find attribute values (and the nodes associated with them) at the same given time, which are related in the given way. A relation between graph elements may also be specified.

? g_1, g_2, y_1, y_2:
$\boldsymbol{f}(\mathbf{t}, g_1) = y_1; \boldsymbol{f}(\mathbf{t}, g_2) = y_2;$
$y_1\, \boldsymbol{\Lambda}\, y_2$; $\boldsymbol{g_1}\, \boldsymbol{\Phi}\, g_2$</td></tr>
<tr><td>**Different time**</td><td>Find the attribute value (and associated node) at a given time, which is related in the given way to an attribute value associated with a given graph object at a different given time point.

? g_2, y_1, y_2:
$\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) = y_1; \boldsymbol{f}(\mathbf{t_2}, g_2) = y_2;$
$y_1\, \boldsymbol{\Lambda}\, y_2$;</td><td>Find attribute values (and the nodes associated with them) at two given times, which are related in the given way. A relation between graph elements may also be specified.

? g_1, g_2, y_1, y_2:
$\boldsymbol{f}(\mathbf{t_1}, g_1) = y_1; \boldsymbol{f}(\mathbf{t_2}, g_2) = y_2;$
$y_1\, \boldsymbol{\Lambda}\, y_2$; $\boldsymbol{g_1}\, \boldsymbol{\Phi}\, g_2$</td></tr>
</table>

	Graph elements (nodes, edges, graph objects)			
	Both constraints		One constraint, one target	Both are targets
	Same element	Different elements		
One constraint, one target	Find an attribute value (and the time point at which it occurs) associated with a given graph element, which is related in the given way to an attribute value associated with a the same graph element at a given time. A relation between time points may also be specified. ? t_2, y_1, y_2: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g}) = y_1$; $\boldsymbol{f}(t_2, \mathbf{g}) = y_2$; $y_1 \boldsymbol{\Lambda} y_2$	Find an attribute value (and the time point at which it occurs) associated with a given graph element, which is related in the given way to an attribute value associated with a different given graph element at a given time. A relation between time points may also be specified. ? t_2, y_1, y_2: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(t_2, \mathbf{g_2}) = y_2$; $t_1 \boldsymbol{\Psi} t_2$; $y_1 \boldsymbol{\Lambda} y$	Find an attribute value (and the time point and graph element for which it occurs) related in the given way to an attribute value which is associated with a given graph element at a given time point. Relations between time points and/or graph elements may also be specified. ? t_2, g_2, y_1, y_2: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(t_2, g_2) = y_2$; $y_1 \boldsymbol{\Lambda} y_2$ Or Find attribute values related in the given way where one of the values occurs at a given time, and the other is associated with a given graph element. Also find the unspecified graph element and time point associated with the attribute values. Relations between time points and/or graph elements may also be specified. ? t_2, g_1, y_1, y_2: $\boldsymbol{f}(\mathbf{t_1}, g_1) = y_1$; $\boldsymbol{f}(t_2, \mathbf{g_2}) = y_2$; $t_1 \boldsymbol{\Psi} t_2$; $g_1 \boldsymbol{\Phi} g_2$; $y_1 \boldsymbol{\Lambda} y_2$	Find attribute values related in the given way where one of the values occurs at the given time. Relations between time points and graph elements may also be specified. t_2, g_1, g_2, y_1, y_2: $\boldsymbol{f}(\mathbf{t_1}, g_1) = y_1$; $\boldsymbol{f}(t_2, g_2) = y_2$; $\mathbf{t_1} \boldsymbol{\Psi} t_2$; $g_1 \boldsymbol{\Phi} g_2$; $y_1 \boldsymbol{\Lambda} y_2$

	Graph elements (nodes, edges, graph objects)			
	Both constraints		**One constraint, one target**	**Both are targets**
	Same element	**Different elements**		
Both are targets	Find attribute values (and the time points at which they occur) associated with the same given graph element, which are related in the given way. A relation between time points may also be specified. ? t_1, t_2, y_1, y_2: $\boldsymbol{f}(t_1, \mathbf{g}) = y_1$; $\boldsymbol{f}(t_2, \mathbf{g}) = y_2$; $y_1 \boldsymbol{\Lambda} y_2$; $t_1 \boldsymbol{\Psi} t_2$;	Find attribute values (and the time points at which they occur) associated with two given graph elements, which are related in the given way. A relation between time points may also be specified. ? t_1, t_2, y_1, y_2: $\boldsymbol{f}(t_1, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(t_2, \mathbf{g_2}) = y_2$; $t_1 \boldsymbol{\Psi} t_2$; $y_1 \boldsymbol{\Lambda} y_2$	Find attribute values related in the given way, where one of the attribute values is associated with a given graph element. Relations between time points and/or graph elements may also be specified. ? t_1, t_2, g_2, y_1, y_2: $\boldsymbol{f}(t_1, \mathbf{g_1}) = y_1$; $\boldsymbol{f}(t_2, g_2) = y_2$; $t_1 \boldsymbol{\Psi} t_2$; $g_1 \boldsymbol{\Phi} g_2$; $y_1 \boldsymbol{\Lambda} y_2$	Find attribute values related in the given way. Relations between time points and/or graph elements may also be specified. ? $t_1, t_2, g_1, g_2, y_1, y_2$: $\boldsymbol{f}(t_1, g_1) = y_1$; $\boldsymbol{f}(t_2, g_2) = y_2$; $t_1 \boldsymbol{\Psi} t_2$; $g_1 \boldsymbol{\Phi} g_2$; $y_1 \boldsymbol{\Lambda} y_2$

Figure 106 Relation seeking, quadrant 1: considers elementary relation seeking involving graph elements (nodes, edges, graph objects) and time points (i.e. the elementary comparison tasks)

			Graph subsets			
			Both constraints		One constraint, one target	Both are targets
			Same subset	Different subsets		
Time points	Both constraints	Same time	Not applicable		Find a pattern and the graph subset over which it occurs at a given time point, which is related in the given way to a pattern over a given graph subset at the same time point. A relation between graph subsets may also be specified. ? G'', P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t}) \approx P_2$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1\ \boldsymbol{\Lambda}\ P_2$	Find patterns related in the given way at the same time point. A relation between graph subsets may also be specified. ? G', G'', P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t}) \approx P_2$; $G'\ \boldsymbol{\Phi}\ G''$; $P_1\ \boldsymbol{\Lambda}\ P_2$
		Different times			Tasks as above, but involving two different time points. ? G'', P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t_2}) \approx P_2$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1\ \boldsymbol{\Lambda}\ P_2$	Tasks as above, but involving two different time points. ? G', G'', P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t_2}) \approx P_2$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1\ \boldsymbol{\Lambda}\ P_2$

	Graph subsets			
	Both constraints		One constraint, one target	Both are targets
	Same subset	Different subsets		
One constraint, one target	Find a pattern (and time point) associated with a given graph subset, which is related in the given way to a pattern associated with the same graph subset at a given time. A relation between time points may also be specified. ? t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_2) \approx P_2$; $\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$; $P_1 \wedge P_2$	Find a pattern (and time point) associated with a given graph subset, which is related in the given way to a pattern associated with a different given graph subset at a given time. A relation between time points may also be specified. ? t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''}, x_2 = t_2) \approx P_2$; $\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$; $P_1 \wedge P_2$	Find a pattern and the time point and graph subset over which it occurs related in the given way to a pattern associated with a given graph subset at a given time point. Relations between time points and graph subsets may also be specified. ? G'', t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$; $\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1 \wedge P_2$ Or Find patterns related in the given way where one of the patterns occurs at a given time, and the other occurs over a given graph subset. Also find the unspecified graph subset and time point over which/at the patterns occur. Relations between time points and graph subsets may also be specified. ? = G'', t_1, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_1) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = \mathbf{t_2}) \approx P_2$; $t_1\ \boldsymbol{\Psi}\ \mathbf{t_2}$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1 \wedge P_2$	Find patterns related in the given way where one of the patterns occurs at the given time. Relations between time points and graph subsets may also be specified. ? G', G'', t_1, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = \mathbf{t_1}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$; $\mathbf{t_1}\ \boldsymbol{\Psi}\ t_2$; $G'\ \boldsymbol{\Phi}\ G''$; $P_1 \wedge P_2$

	Graph subsets			
	Both constraints		One constraint, one target	Both are targets
	Same subset	Different subsets		
Both are targets	Find patterns (and the time points at which they occur) associated with a single given graph subset, which are related in the given way. A relation between time points may also be specified. ? t_1, t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_1) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_2) \approx P_2$; $t_1\, \boldsymbol{\Psi}\, t_2$; $P_1 \boldsymbol{\wedge} P_2$	Find patterns (and the time points at which they occur) associated with two given graph subsets, which are related in the given way. A relation between time points may also be specified. ? t_1, t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_1) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''}, x_2 = t_2) \approx P_2$; $t_1\, \boldsymbol{\Psi}\, t_2$; $P_1 \boldsymbol{\wedge} P_2$	Find patterns related in the given way, where one of the patterns is associated with a given graph subset. Relations between time points and graph subsets may also be specified. ? G'', t_1, t_2, P_1, P_2 : $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'}, x_2 = t_1) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$; $t_1\, \boldsymbol{\Psi}\, t_2$; $\mathbf{G'}\, \boldsymbol{\Phi}\, G''$; $P_1 \boldsymbol{\wedge} P_2$	Find patterns related in the given way. Relations between time points and graph subsets may also be specified. ? $G', G'', t_1, t_2, P_1, P_2$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G', x_2 = t_1) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'', x_2 = t_2) \approx P_2$; $t_1\, \boldsymbol{\Psi}\, t_2$; $G'\, \boldsymbol{\Phi}\, G''$; $P_1 \boldsymbol{\wedge} P_2$

Figure 107 Relation seeking quadrant 2: considers synoptic relation seeking involving the behaviour of an attribute over the graph (or a graph subset)

			Graph elements (nodes, edges, graph objects)			
			Both constraints		**One constraint, one target**	**Both are targets**
			Same element	**Different elements**		
Time intervals	**Both constraints**	**Same time**	Not applicable		Find a pattern (and the graph element associated with it) which occurs over a given time interval and is related in the given way to a pattern associated with a given graph element over the same time interval. A relation between graph elements may also be specified.[52] ? g_2, P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx P_2$; $P_1 \mathbf{\Lambda} P_2$; $\mathbf{g_1} \Phi g_2$	Find patterns (and their associated graph elements) which occur over the same given time interval and are related in the given way. A relation between graph elements may also be specified. ? g_1, g_2, P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx P_2$; $P_1 \mathbf{\Lambda} P_2$; $g_1 \Phi g_2$
		Different times			Find a pattern (and the graph element associated with it) which occurs over a given time interval and is related in the given way to a pattern associated with a given graph element over a given time interval. A relation between graph elements may also be specified ? g_2, P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T'}) \approx P_2$; $P_1 \mathbf{\Lambda} P_2$; $\mathbf{g_1} \Phi g_2$	Find patterns (and their associated graph elements) which occur over two given time intervals and are related in the given way. A relation between graph elements may also be specified ? g_1, g_2, P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T''}) \approx P_2$; $P_1 \mathbf{\Lambda} P_2$; $g_1 \Phi g_2$

[52] In all cases in this table, if we wish to specify a linking relation between the graph elements, we must also specify a time at which the linking relation occurs i.e. ($\mathbf{g_1}$, t) Φ (g_2, t): *'a given linking relation exists between the graph elements at time* **t***'.*

	Graph elements (nodes, edges, graph objects)			
	Both constraints		One constraint, one target	Both are targets
	Same element	Different elements		
One constraint, one target	Find a pattern (and the time interval over which it occurs) for a given graph element, which is related in the given way to a pattern associated with the same graph element over a given time interval. A relation between time intervals may also be specified. ? T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g}, x_2 \in T'') \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \wedge P_2$	Find a pattern (and the time interval over which it occurs) for a given graph element, which is related in the given way to a pattern associated with a given graph element over a given time interval. A relation between time intervals may also be specified. ? T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_2}, x_2 \in T'') \approx P_2$; $T'\ \boldsymbol{\Psi}\ T''$; $P_1 \wedge P_2$	Find a pattern, and the graph element and time interval over which it occurs, which is related in the given way to a pattern associated with a given graph element over a given time interval. A relation between time intervals and/or graph elements may also be specified. ? g_2, T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \wedge P_2$; $\mathbf{g_1}\ \Phi\ g_2$ Or Find patterns related in the given way where one of the patterns occurs over a given time interval, and the other is associated with a given graph element. Also find the unspecified graph element and time interval associated with the patterns). A relation between time intervals and/or graph elements may also be specified. ? g_2, T', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in T') \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in \mathbf{T''}) \approx P_2$; $T'\ \boldsymbol{\Psi}\ \mathbf{T''}$; $P_1 \wedge P_2$; $\mathbf{g_1}\ \Phi\ g_2$	Find patterns related in the given way where one of the patterns occurs over a given time interval. A relation between time intervals and/or graph elements may also be specified. ? g_1, g_2, T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in \mathbf{T'}) \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \wedge P_2$; $g_1\ \Phi\ g_2$

	Graph elements (nodes, edges, graph objects)			
	Both constraints		One constraint, one target	Both are targets
	Same element	Different elements		
Both are targets	Find patterns (and the time intervals over which they occur) associated with a single graph element, which are related in the given way. A relation between time intervals may also be specified. ? T', T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g}, x_2 \in T') \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g}, x_2 \in T'') \approx P_2$; $T' \boldsymbol{\Psi} T''$; $P_1 \boldsymbol{\Lambda} P_2$	Find patterns (and the time intervals over which they occur) associated with two given graph elements, which are related in the given way. A relation between time intervals may also be specified. ? T', T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g_1}, x_2 \in T') \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \boldsymbol{g_2}, x_2 \in T'') \approx P_2$; $T' \boldsymbol{\Psi} T''$; $P_1 \boldsymbol{\Lambda} P_2$	Find patterns related in the given way, where one of the patterns is associated with a given graph element. A relation between time intervals and/or graph elements may also be specified. g_2, T', T'', P_1, P_2: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = \mathbf{g_1}, x_2 \in T') \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx P_2$; $T' \boldsymbol{\Psi} T''$; $P_1 \boldsymbol{\Lambda} P_2$; $\mathbf{g_1} \Phi g_2$	Find patterns related in the given way. A relation between time intervals and/or graph elements may also be specified. ? $g_1, g_2, T', T'', P_1, P_2$: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_1, x_2 \in T') \approx P_1$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 = g_2, x_2 \in T'') \approx P_2$; $T' \boldsymbol{\Psi} T''$; $P_1 \boldsymbol{\Lambda} P_2$; $g_1 \Phi g_2$

Figure 108 Relation seeking quadrant 3: considers relation seeking tasks involving the behaviour of an attribute of a single graph element over time (i.e. a temporal trend)

<table>
<tr><td colspan="3"></td><th colspan="4">Graph subsets</th></tr>
<tr><td colspan="3"></td><th colspan="2">Both constraints</th><th rowspan="2">One constraint, one target</th><th rowspan="2">Both are targets</th></tr>
<tr><td colspan="3"></td><th>Same subset</th><th>Different subsets</th></tr>
<tr><th rowspan="2">Time intervals</th><th rowspan="2">Both constraints</th><th>Same time</th><td colspan="2" rowspan="2">Not applicable</td><td>Find a pattern (and the graph subset associated with it) which is associated with a given time interval and is related in the given way to a pattern associated with a given graph subset and the same time interval. A relation between graph subsets may also be specified.

? G'', P_1, P_2:
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx P_1$;
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G''\} \approx P_2$;
$\mathbf{G'}\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$

or

? G'', P_1, P_2:
$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T'}\} \approx P_2$;
$\mathbf{G'}\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$</td><td>Find patterns (and their associated graph subsets) which are associated with a single given time interval and are related in the given way. A relation between graph subsets may also be specified.

? G', G'', P_1, P_2:
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx P_1$;
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G''\} \approx P_2$;
$G'\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$

or

? G', G'', P_1, P_2:
$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx P_1$;
$\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T'}\} \approx P_2$;
$G'\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$</td></tr>
<tr><th>Different times</th><td>Find a pattern (and the graph subset associated with it) which is associated with a given time interval and is related in the given way to a pattern associated with a given graph subset and a given time interval. A relation between graph subsets may also be specified.

? G'', P_1, P_2:
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx P_1$;
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in G''\} \approx P_2$;
$\mathbf{G'}\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$</td><td>Find patterns (and their associated graph subsets) which are associated with two given time intervals and are related in the given way. A relation between graph subsets may also be specified.

? G', G'', P_1, P_2:
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx P_1$;
$\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T''})] \mid x_1 \in G''\} \approx P_2$;
$G'\ \boldsymbol{\Phi}\ G''$;
$P_1 \boldsymbol{\wedge} P_2$

or</td></tr>
</table>

	Graph subsets			
	Both constraints		One constraint, one target	Both are targets
	Same subset	Different subsets		
			or ? G'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx P_2$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$	? G', G'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx P_2$; $G'\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$
One constraint, one target	Find a pattern (and the time interval with which it is associated) for a given graph subset, which is related in the given way to a pattern associated with the same graph subset and a given time interval. A relation between time intervals may also be specified. ? T'', P_1, P_2: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx P_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G'}\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \boldsymbol{\wedge} P_2$ or ? T'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T''\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \boldsymbol{\wedge} P_2$	Find patterns (and the time intervals over which they occur) associated with two given graph subsets, which are related in the given way. A relation between time intervals may also be specified. ? T'', P_1, P_2: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx P_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $P_1 \boldsymbol{\wedge} P_2$ or ? T'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in T''\} \approx P_2$; $T'\ \boldsymbol{\Psi}\ T''$; $P_1 \boldsymbol{\wedge} P_2$	Find a pattern, and the graph subset and time interval with which it is associated, which is related in the given way to a pattern associated with a given graph subset and a given time interval. Relations between time intervals and/or graph subsets may also be specified. ? G'', T'', P_1, P_2: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in \mathbf{G'}\} \approx P_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$ or ? G'', T'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $\mathbf{G'}\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$ OR	Find patterns related in the given way where one of the patterns involves a given time interval. Relations between time intervals and/or graph subsets may also be specified. ? G', G'', T'', P_1, P_2: $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx P_1$; $\boldsymbol{\beta}_G\{\boldsymbol{\beta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $G'\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$ or ? G', G'', T'', P_1, P_2: $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in \mathbf{T'}\} \approx P_1$; $\boldsymbol{\beta}_T\{\boldsymbol{\beta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx P_2$; $\mathbf{T'}\ \boldsymbol{\Psi}\ T''$; $G'\ \boldsymbol{\Phi}\ G''$; $P_1 \boldsymbol{\wedge} P_2$

	Graph subsets			
	Both constraints		One constraint, one target	Both are targets
	Same subset	Different subsets		
			Find patterns related in the given way where one of the patterns is associated with a given time interval, and the other is associated with a given graph subset. Also find the unspecified graph subset and time interval associated with the patterns). Relations between time intervals and/or graph subsets may also be specified. $? G', T'', P_1, P_2$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in \mathbf{T'})] \mid x_1 \in G'\} \approx P_1$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx P_2$; $T' \boldsymbol{\Psi} \mathbf{T''}$; $\mathbf{G'} \boldsymbol{\Phi} G''$; $P_1 \boldsymbol{\Lambda} P_2$ or $? G'', T', T'', P_1, P_2$: $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx P_1$; $\boldsymbol{\theta}_T\{\boldsymbol{\theta}_G[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in \mathbf{T''}\} \approx P_2$; $T' \boldsymbol{\Psi} \mathbf{T''}$; $\mathbf{G'} \boldsymbol{\Phi} G''$; $P_1 \boldsymbol{\Lambda} P_2$	
Both are targets	Find patterns (and the time intervals over which they occur) associated with the same given graph subset, which are related in the given way. A relation between time intervals may also be specified. $? T', T'', P_1, P_2$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx P_1$;	Find patterns (and the time intervals over which they occur) associated with two given graph subsets, which are related in the given way. A relation between time intervals may also be specified. $? T', T'', P_1, P_2$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx P_1$;	Find patterns related in the given way, where one pattern is associated with a given graph subset. Relations between time intervals and/or graph subsets may also be specified. $? G'', T', T'', P_1, P_2$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in \mathbf{G'}\} \approx P_1$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx P_2$; $T' \boldsymbol{\Psi} T''$;	Find patterns related in the given way. Relations between time intervals and/or graph subsets may also be specified. $? G', G'', T', T'', P_1, P_2$: $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T')] \mid x_1 \in G'\} \approx P_1$; $\boldsymbol{\theta}_G\{\boldsymbol{\theta}_T[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in G''\} \approx P_2$; $T' \boldsymbol{\Psi} T''$; $G' \boldsymbol{\Phi} G''$;

		Graph subsets			
		Both constraints		One constraint, one target	Both are targets
		Same subset	Different subsets		
		$\boldsymbol{\beta_G}\{\boldsymbol{\beta_T}[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G'}\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $P_1 \wedge P_2$ or ? T', T'', P_1, P_2: $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx P_1$; $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T''\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $P_1 \wedge P_2$	$\boldsymbol{\beta_G}\{\boldsymbol{\beta_T}[\boldsymbol{f}(x_1, x_2) \mid x_2 \in T'')] \mid x_1 \in \mathbf{G''}\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $P_1 \wedge P_2$ or ? T', T'', P_1, P_2: $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G'})] \mid x_2 \in T'\} \approx P_1$; $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''})] \mid x_2 \in T''\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $P_1 \wedge P_2$	$\mathbf{G'} \,\boldsymbol{\Phi}\, G''$; $P_1 \wedge P_2$ or ? G'', T', T'', P_1, P_2: $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in \boldsymbol{G'})] \mid x_2 \in T'\} \approx P_1$; $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $\mathbf{G'} \,\boldsymbol{\Phi}\, G''$; $P_1 \wedge P_2$	$P_1 \wedge P_2$ or ? $G', G'', T', T'', P_1, P_2$: $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G')] \mid x_2 \in T'\} \approx P_1$; $\boldsymbol{\beta_T}\{\boldsymbol{\beta_G}[\boldsymbol{f}(x_1, x_2) \mid x_1 \in G'')] \mid x_2 \in T''\} \approx P_2$; $T' \,\boldsymbol{\Psi}\, T''$; $G' \,\boldsymbol{\Phi}\, G''$; $P_1 \wedge P_2$

Figure 109 Relation seeking quadrant 4: considers relation seeking tasks involving aspectual behaviours (i) the behaviour of temporal trends for all graph elements, over the graph (ii) the behaviour of an attribute over the graph, over time

Appendix C **Study Part 1 – Instructions to Participants**

Assessing the utility of a taxonomic approach to requirements gathering in visualisation

Background to the study

When developing a visualisation system, it is important to understand what questions a person who will use the system would like to be able to ask of the data. We would like to develop a visualisation system to help **better understand collaborative working practices and publishing rates** in the School of Computing. We therefore would like to find out what questions people using the visualisation system would like to ask of the data that we have available.

One way to help understand collaborative working practices is to construct a co-authorship network showing who co-authors with whom. In such a network, authors are connected to one another according to whether they have published together. These networks may change over time with new authors joining the network and others leaving the network. Co-authoring within the network may also change: authors may publish repeatedly with the same colleagues or collaborate with different authors at different times.

In addition to considering the network structure and how it changes over time, we might also consider publishing rates in this network context – perhaps there is some relationship between the network structure (collaborative working practices) and the amount which individuals publish? The number of publications is also likely to vary over time, with authors publishing more or less frequently in certain years.

While there may be many outside factors affecting publication rates and co-authorship (teaching loads, ease or difficulty of publishing within a given research area, etc.), as a first step, we would like to use visualisation techniques to gain a basic understanding of what publishing rates and co-authorship look like within the School and how this has changed over the past three decades.

With this in mind, in order to inform the design of the visualisation tool, we would like you to help by suggesting specific questions relating to the co-authorship network and publishing rates that it might be interesting to ask of the data which we have available to us, which is described below.

Data

The School holds a large amount of data relating to the publications of its members of staff. Each member of staff (an author) has a list of publications and belongs to a research centre. For the purposes of this study, we have access to the following metadata associated with authors and publications for use in our visualisation system:

Authors:

- Name
- Research Centre (CAVES, CCER, CDCNS, CID, CSI)
- Joining and leaving dates

Publications:
- The list of authors
- The year in which it was published
- The type of publication (conference proceeding, journal article, book chapter, etc.)

To illustrate, an extract of the data is included in Tables 1 and 2, below. The full dataset can be found at: https://intranet.institute.napier.ac.uk/iidi/queries

Table 45 Authors

Name	Research Centre	Joined	Left
Alan Cannon	CAVES	2003	-
Kevin Chalmers	CAVES	2005	-
Paul Craig	CAVES	2008	2012
Martin Graham	CAVES	1998	2015
Jessie Kennedy	CAVES	1991	-
Natalie Kerracher	CAVES	2010	-
Robert Kukla	CAVES	1996	-
Paul Shaw	CAVES	2008	-
Alistair Thomson	CAVES	2012	2013
…	…	…	…

Table 46 Publications

ID	Year	Authors	Type
1456	2015	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers	Journal Article
1455	2015	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers, Martin Graham	Conference Paper
1444	2014	Jessie Kennedy, *Externals*	Book Chapter
1401	2014	Martin Graham, Jessie Kennedy	Journal Article
1385	2014	Natalie Kerracher, Jessie Kennedy, Kevin Chalmers	Conference Paper
1343	2014	Jessie Kennedy , *Externals*	Journal Article
1341	2014	Paul Shaw, Martin Graham, Jessie Kennedy, *External*	Journal Article
1248	2013	Paul Craig, Alan Cannon, Robert Kukla, Jessie Kennedy	Journal Article
1219	2013	Jessie Kennedy, Martin Graham, *Externals*	Conference Paper
1107	2013	Alistair Thomson, Martin Graham, Jessie Kennedy	Conference Paper
…	…	…	…

Table 3 Authors' Publication Counts Over Time

Author	Year	Publication Count
Kevin Chalmers	2015	2
Kevin Chalmers	2014	8
Kevin Chalmers	2013	8
…	…	…
Jessie Kennedy	2015	2
Jessie Kennedy	2014	5
Jessie Kennedy	2013	3
…	…	…

From this data, we can extract a co-authorship network where authors are connected according to whether they have published together. For example, in 2015, Jessie Kennedy is connected to Natalie Kerracher, Kevin Chalmers, and Martin Graham.

The full dataset contains data on approximately two-hundred authors and nearly two thousand publications. It spans a period of over thirty years, during which time authors have joined and left the network, and published varying amounts and types of publications each year. We can therefore construct a large co-authorship network which changes over time, in terms of who belongs to the network, who is publishing with whom in each year, and the amount and type of publications being published.

Part 1

(i) In what capacity might this data set be of interest to you? (Please check all which are relevant):

- ☐ In a management capacity
- ☐ Understanding my own data, e.g. looking at my own publishing track record, comparing myself with colleagues etc.
- ☐ Finding potential collaborators
- ☐ Understanding the data relating to my research group
- ☐ Other (please specify):

(ii) Consider the dataset described above. If you were to try and understand the publishing rates and co-authoring behaviour within the School over the years, what questions might you want to ask of this dataset? Please spend around 10-15 minutes considering the data, and make a note of any questions which might be of interest to you in the table below. ***Please include only questions that it would be possible to answer from the available data as described above (for example, this particular dataset does not have data on research topics or publishing venues).*** Please be aware that there are no right or wrong answers – all responses will be useful for the purposes of the study.

If additional space is required, please use a separate sheet. Please rate your questions on a scale of 1-4 in terms of how interesting they are to you, using the following scale:

1 = slightly interesting
2 = moderately interesting
3 = very interesting
4 = extremely interesting

Question	**Rating (1-4)**

Thank you for completing part 1 of this study. Please return your completed answers and list of questions to Natalie Kerracher (n.kerracher@napier.ac.uk; room C40 Merchiston) by ****Friday 12th February**.**

Appendix D Study Part 2 – Instructions to Participants

Instructions

When developing a visualisation system, it is important to understand what questions a person who will use the system would like to be able to ask of the data. We would like to develop a visualisation system to help **better understand collaborative working practices and publishing rates in the School of Computing**. We therefore would like to find out what questions people using the visualisation system would like to ask of the data that we have available.

For this part of the study, we have provided a list of questions covering different aspects of the data. Please rate each question on a scale of 0-4 in terms of how interesting they are to you, using the following scale:

0 = of no interest
1 = slightly interesting
2 = moderately interesting
3 = very interesting
4 = extremely interesting

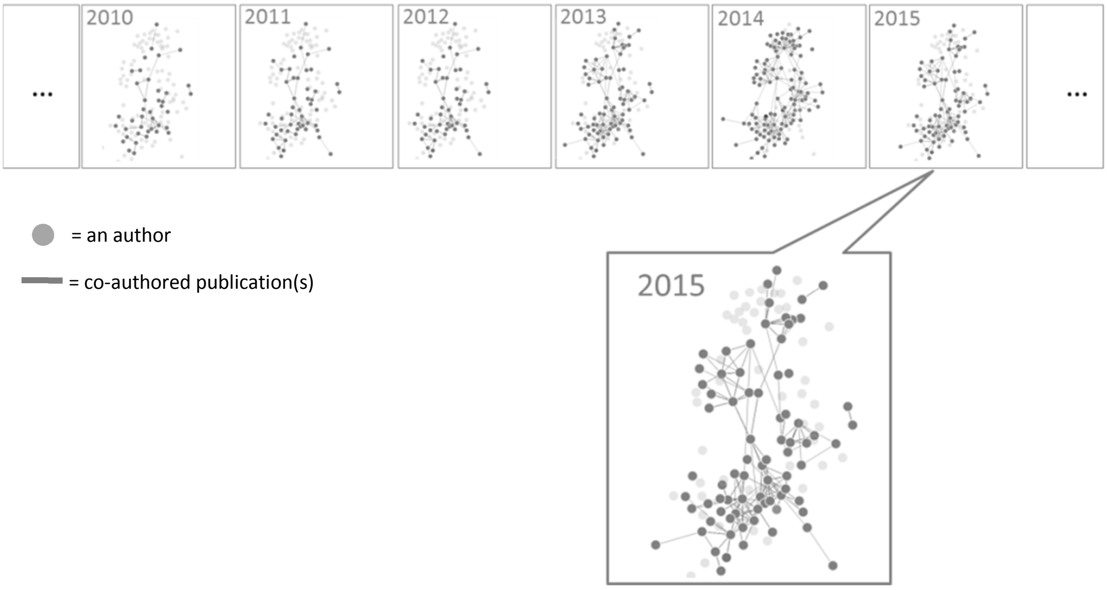
Please put your answers in the boxes marked **[Your Rating:]**

If you do not understand a question, please feel free to contact me for clarification (room C40; n.kerracher@napier.ac.uk; ext 2798). Otherwise, please simply note **DNU** (do not understand) in the relevant box. If you have any comments on the questions, please feel free to note them and return them to me along with your completed form, if at all possible, by **Friday 21 October.**

Please note that images (charts, networks etc.) are used to help illustrate the question only and are constructed using synthetic data. There may be other, more appropriate ways to visualise the data when answering a particular question.

Questions

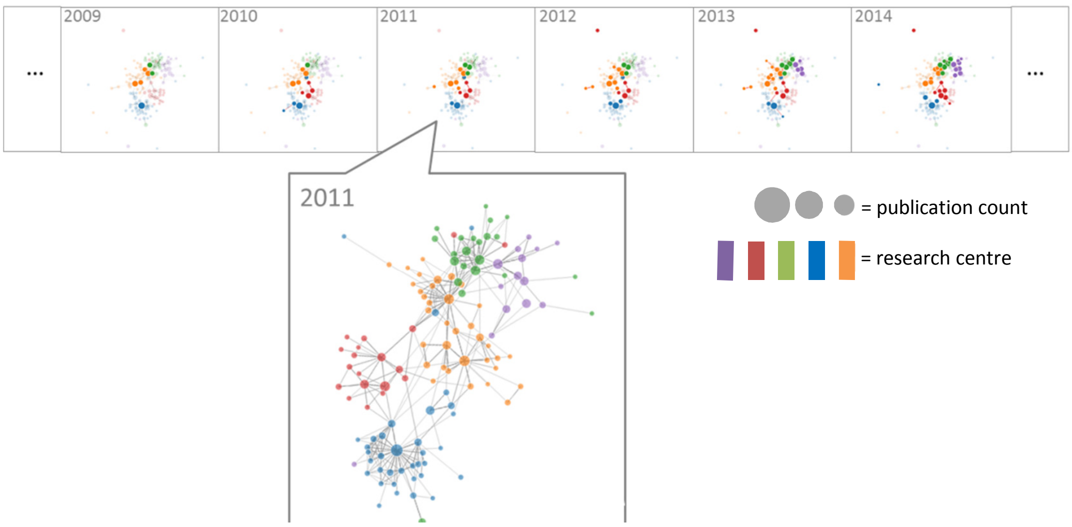
1. Are you interested in understanding the co-authorship network (or part of the network) in **a single year**…



Structure of the co-authorship network in 2015

I. …in terms of its structure? E.g. *How big is the network? Are there any interesting patterns of co-authorship? Is the network tightly or sparsely connected (i.e. lots or little co-authorship)? Is the network completely connected or fragmented into smaller co-authoring groups? Are there any authors who don't co-author?*

[Your Rating:]



Distribution of publication counts (circle size) and research centres (colour) over the network in 2011

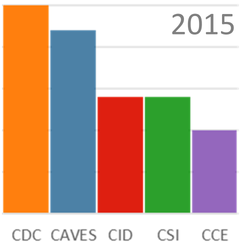
II. …in terms of the network's structure and distribution of publication counts? E.g. *Do more collaborative authors have higher publication counts? What about non-collaborative authors – do they have high or low numbers of publications? Are there any groups of co-authors with particularly high publication counts?*

[Your Rating:]

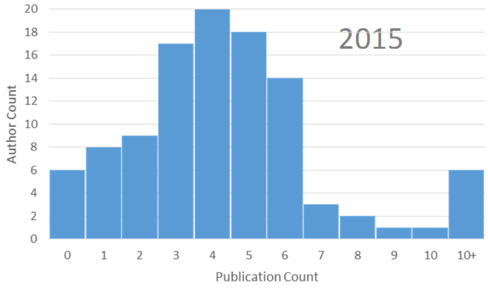
III. …in terms of the network's structure and distribution of research centres? E.g. *Do authors from the same research centre tend to publish together or with authors from different research centres? What does co-authorship in a particular research centre look like?*

[Your Rating:]

IV. Would it be interesting to understand frequency distributions or ranking patterns for **a single year**? e.g.



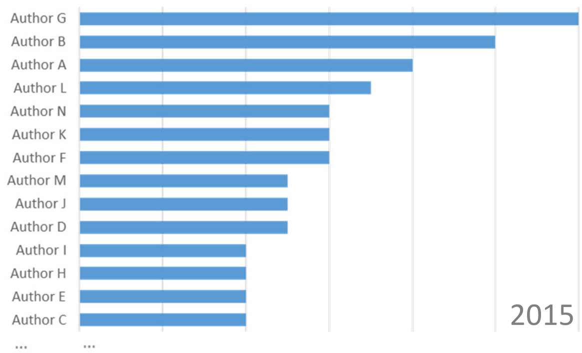
Number of authors in each research centre in 2015



Number of authors by publication counts

a. Frequencies: *the number of authors in each research centre; the number of authors with 0, 1, 2, 3,…, n publications*

[Your Rating:]

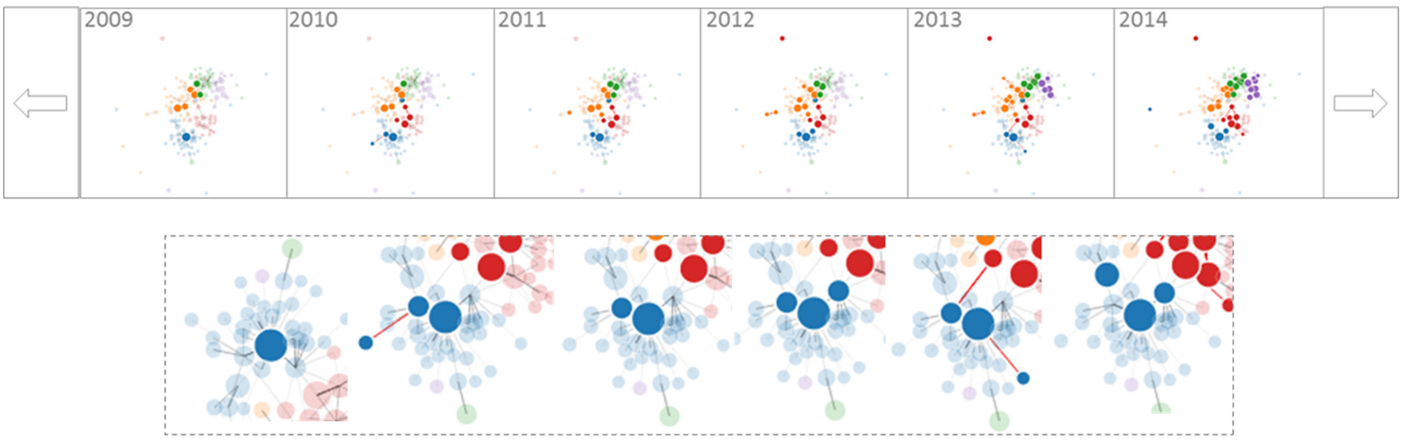


Authors ranked by number of publications in 2015

b. Rankings: *ranking of authors by number of publications/number of each type of publication*

[Your Rating:]

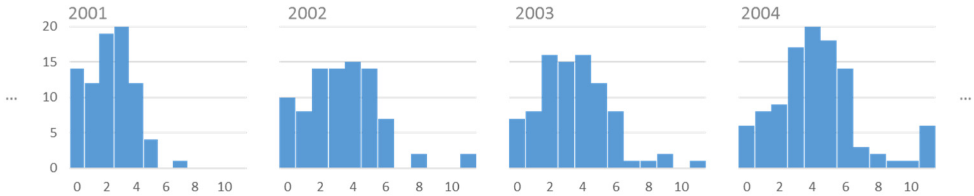
2.



I. Would it be interesting to understand how the network's structure and publication counts change over time? Or how the network's structure and research centre affiliations evolve over time?
E.g. *How does the distribution of publication counts change as the network evolves? Are there any interesting patterns? Do authors with many co-authors have consistently higher numbers of publications over time? What about authors who continuously publish within the same co-author groups – is there a pattern to their amounts or types of publication? Is co-authorship between research centres changing over time?*

[Your Rating:]

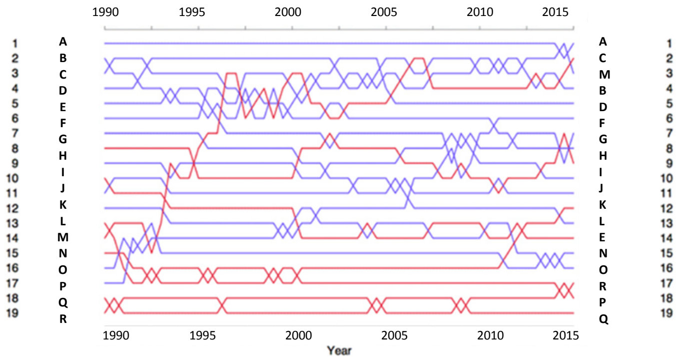
II. Are you interested in understanding how the network changes over time in terms of frequency distributions or ranking patterns? E.g.



Changes in number of authors by publication count, 2001-04

a. Changes in frequency distributions - *How do frequency distributions (e.g. the number of authors in each research centre; the number of authors with 1, 2, 3, …, n publications) change over time?*

[Your Rating:]

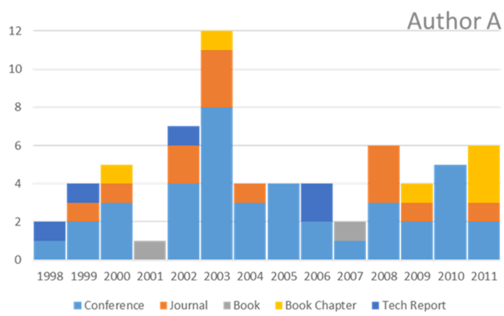


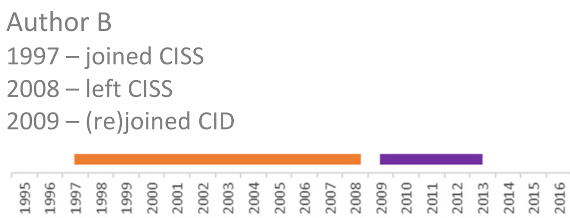
Changes in author rankings by publication count over time

b. Changes in rankings - *how do rankings of authors by number of publications change over time?*

[Your Rating:]

3.

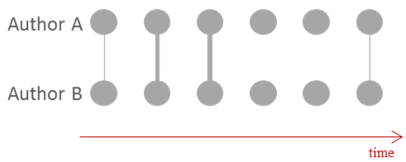


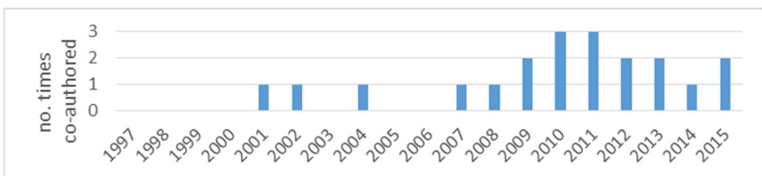


Author B's research centre affiliation over time

Author A's publications over time

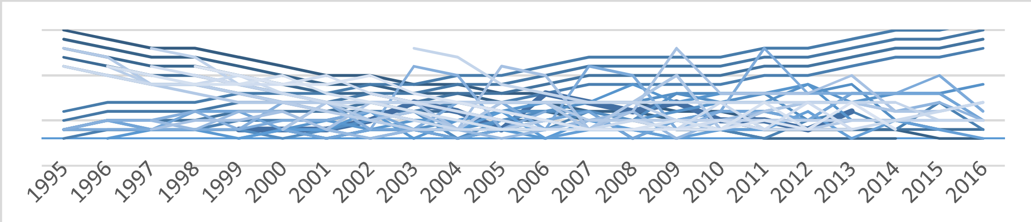
For each individual author, we can look at: how their publication counts and types of publications have changed over time; when they joined and left the School; which research centre they belong to; and whether they moved research centres during this time (see figure, above).





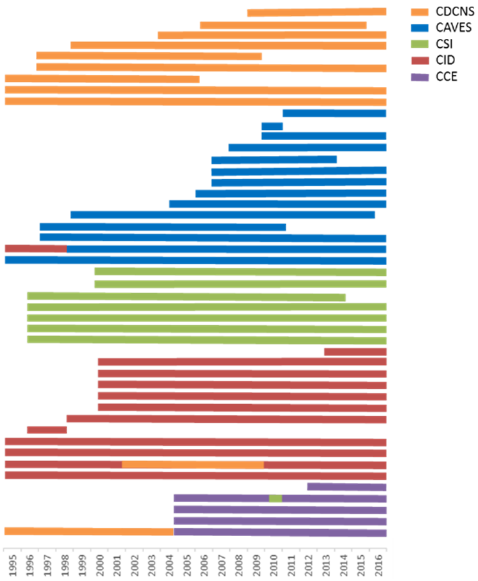
Co-authoring between Author A and Author B over time

We can also look at co-authoring between individual pairs of authors in terms of the amounts and frequency of co-publication over time.



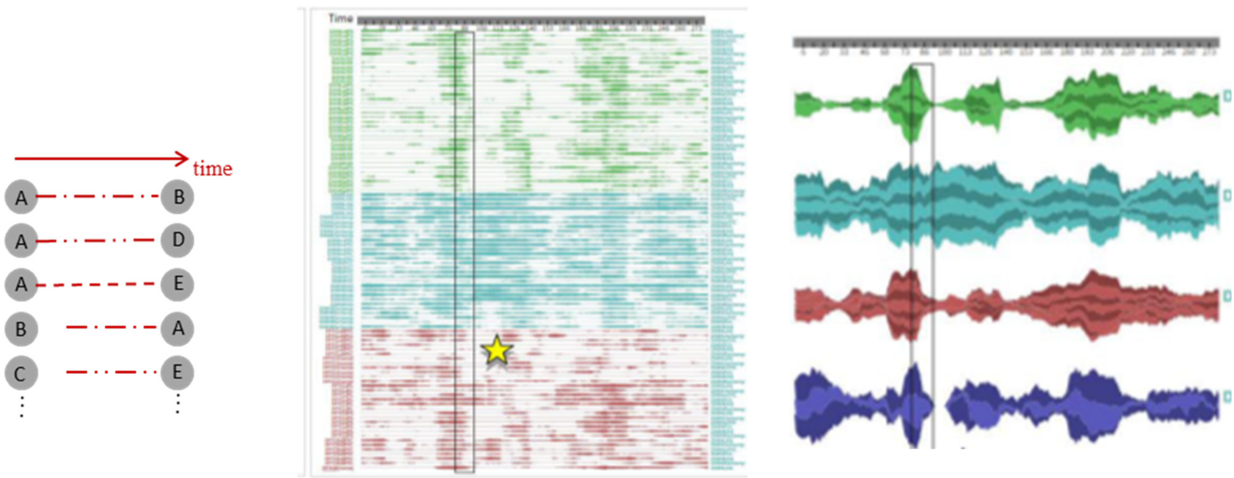
Publication count over time (all authors)

I. Would it be interesting to explore the **set** of trends in publication counts over time, to see if there are any wider patterns within the School? *e.g. Are there general trends in publication amounts (e.g. peaks corresponding to REF dates or management changes)? Are there groups of authors whose publication counts are significantly increasing or decreasing over time?* **[Your Rating:]**



Research centre affiliation over time (all authors)

II. Would it be interesting to explore the research centres to which staff belong and their starting and leaving dates to look for wider patterns within the School? *e.g. How common is it for staff to move research centre? Are there any peaks or troughs in recruitment or leaving, or periods of high movement between research centres?* **[Your Rating:]**

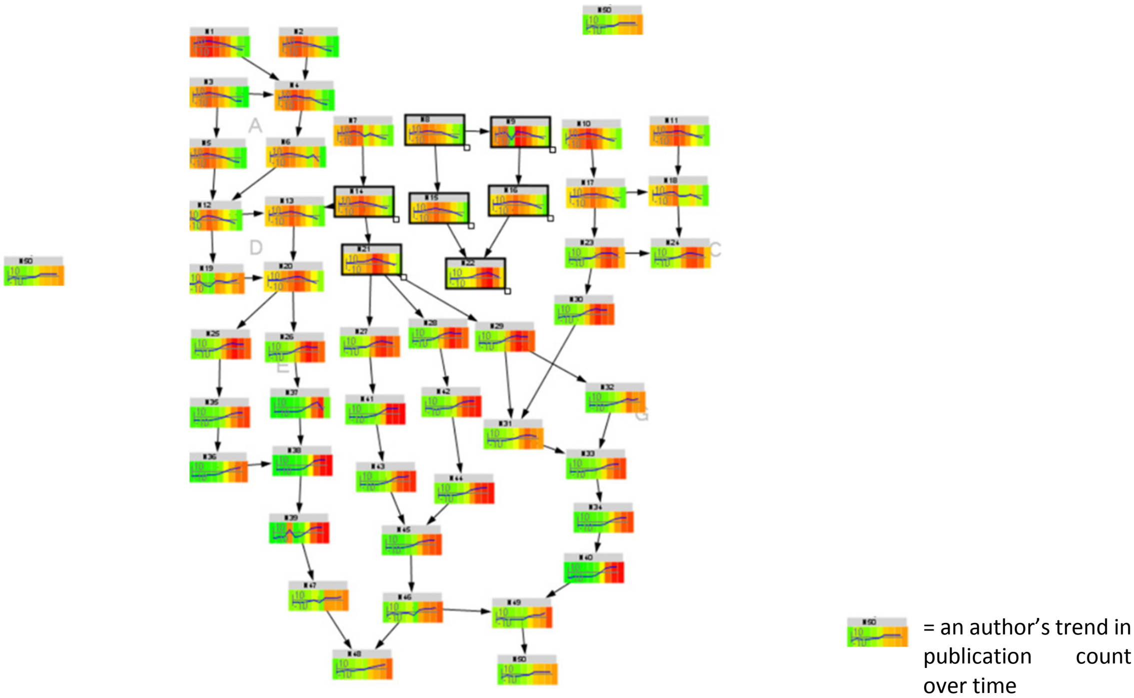


Co-authoring over time for all pairs of authors. Each line represents co-authoring over time between a pair of authors (left and middle). Right: groups of trends.

III. Would it be interesting to look at the trends in co-authorship over time between all pairs of authors e.g. *whether the school is generally becoming more or less collaborative, whether there are particular time* *periods where co-authoring is low or high, or whether the patterns can be grouped into different categories (e.g.by type of collaboration - continuous co-authors, one-off co-authors, intermittent co-authors etc.)*

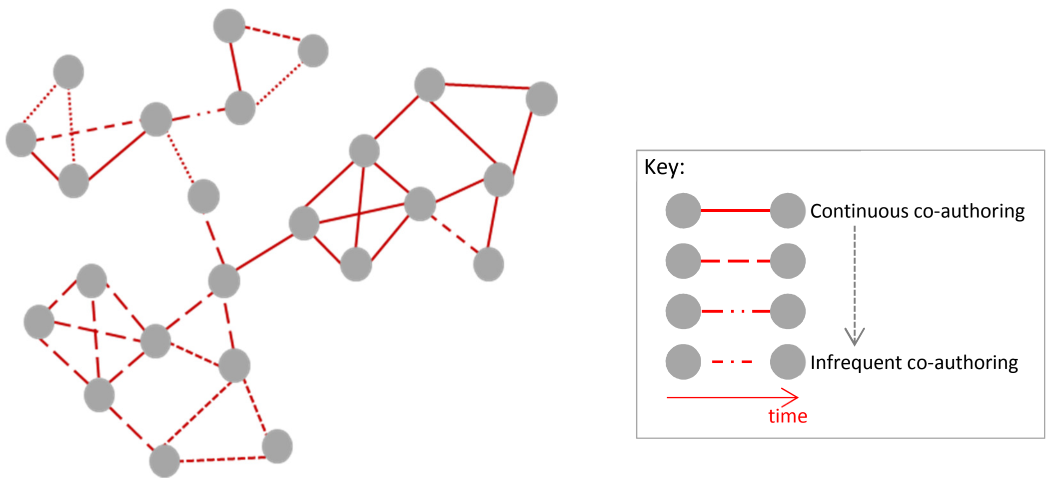
[Your Rating:]

4.



Distribution of trends in publication count over the network

I. Still thinking about individual trends over time, would it be interesting to see how publication counts over time are distributed over the network? e.g. *do groups of authors connected to one another in the network (i.e. collaborators) have similar trends in publication count? Do trends in publication counts over time differ depending on the number of co-authors someone has?* **[Your Rating:]**

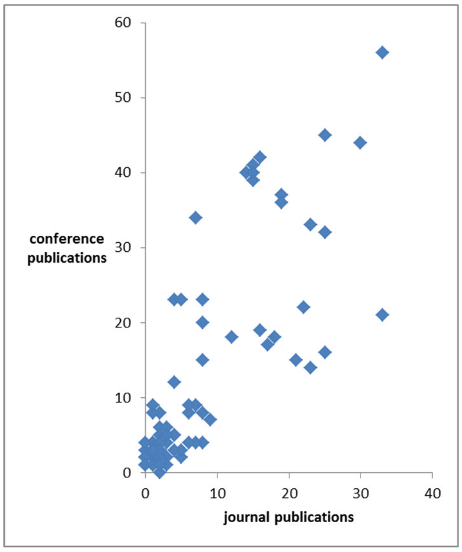


Distribution of trends in co-authoring over the network

II. Would it be interesting to see how trends in co-authoring are distributed over the network? e.g. *are there clusters of similar temporal trends in co-authoring between pairs of authors over time?*

[Your Rating:]

5.



Scatterplot showing the relationship between journal publication counts and conference publication counts

I. Would it be interesting to investigate the relationships (such as influence/dependence and correlation) between the counts of different types of publications, or publication counts and research centre? E.g.

- *Is there a relationship between the publishing rates of different types of publication e.g. do people who publish many journal articles tend to publish fewer conference papers?*
- *Does the research centre to which an author belongs have any influence on how much they publish?*
- *Do high publication counts during one time period (e.g. a REF period) affect publication counts during later time periods?* **[Your Rating:]**

II. Would it be interesting to investigate the relationships (correlation, influence, dependency) between publication count and network structure, or research centre and network structure? E.g.

- *Is there a relationship between an author's position in the co-authoring network (e.g. central, on the periphery of the network) and their publication count?*
- *Does the research centre to which an author belongs affect their position in the network? (e.g. are CAVES authors more likely to be central or on the periphery?)*
- *Do certain patterns in the distribution of publication counts or research centre affiliation over the network precede particular changes in the networks' structure?*
- *Does the structure of the co-authoring network affect publication counts? (e.g. does a fragmented network result in lower or higher publication counts)* **[Your Rating:]**

III. Would it be interesting to investigate the relationship between the structure of the co-authoring network at different time points? Or whether changes in one part of the network affect other parts of the network? E.g.

- *Can we observe any mechanisms by which co-authoring relationships are formed? E.g. do authors with many co-authors increase their number of co-authors over time? Do authors from the same research centre tend to co-author with one another? Does a particular author or group trigger increased collaboration?*
- *How does co-authoring at one point in time predict likelihood of co-authoring in future? Do authors seek to publish with new co-authors or maintain their already established relationships?*
- *Does the structure of the co-authoring network at one point in time affect the structure at later times?*
- *How do changes in co-authoring in one part of the network affect the rest of the network?*

[Your Rating:]

Appendix E Categorisation of Participants’ tasks

Categorisation of participants’ tasks according to quadrant, task type, and whether they involve attribute only, attribute and graph structure, or graph structure only. First number is task number (corresponding to explanation table), number in brackets is the participant’s interest rating e.g. 61. (3) is task 61 which is rated as 3 (very interesting).

		Direct Lookup/ Behaviour Characterisation	Inverse Lookup/ Pattern Search	Direct Comparison	Inverse Comparison	Relation Seeking
Q1	-	61. (3)	*Auxiliary task for:* 7.(2), 10.(2), 11.(2), 13.(2), 42.(2), 67.(1), 38. (3), 39. (3), 40. (3), 41. (2).			22. (2), 48. (3), 58. (3)
Q2	**Structure**					
	Attribute only					35. (2), 41. (2)
	Structure + Attribute					
Q2 aggregated on graph	**Attribute**	49. (3), 48. (3), 58. (3)				
Q3	**Structure**	55b. (3), 56b. (3)				
	Attribute	36. (4), 45. (1), 46. (3), 59. (3), 52. (3), 53. (3)	27. (1), 28. (1), 33. (3), 32. (3), 54. (3), 17. (3), 18. (2)			
Q3 aggregated on time	**Attribute**	34. (4), 68. (2), 60. (3), 62. (2), 29. (3), 47 (3)	20. (3), 26. (4), 63. (2)	51b. (3)		
Q4i	**Structure**	9.(3), 36. (4), 45. (1), 46. (3), 59. (3)	12.(2), 23. (2), 24. (2), 25. (2) , 57. (3)			
	Attribute only					
	Structure + Attribute					
Q4ii (set of temporal trends)	**Structure**					
	Attribute		43. (1)			
Q4ii (distribution of temporal trends over the network)	**Structure**					
	Attribute		40. (3)			
Q4 aggregated on time	**Structure**	64. (2), 31. (4), 15. (2)	4. (3) , 6. (3), 15. (2)	13. (2)		
	Attribute + structure	29. (3)	2. (2), 3. (2), 21. (2), 8. (2)			
	Attribute only	11. (2), 10. (2)	19. (2), 50. (3)			

Q4 aggregated on graph	**Attribute**	53. (3)	54. (3)			
Q4 aggregated on time and graph	**Attribute**	49. (3), 68. (2), 57. (3), 51a. (3)		42. (2), 44. (1)		

Structural Comparison and Relation Seeking tasks:

	Structural comparison	Structural relation seeking
Q1	7. (2) , 67. (1)	38. (3), 55a. (3), 56a. (3), 61. (3), 39. (3); *Auxiliary task for:* 41.(2)

Connection Discovery:

Relationship between network structure and attributes	1.(2)
Relationship between network structures	
Relationship between attributes	

Notes on categorisation of participants' tasks:

		Participant task	Rating (1-4)	Framework category	Notes
1	P1	Whose publication rates have been affected by someone else arriving or leaving	2	Connection discovery (relationship between network structure and attributes)	
2	P1	Who are the people who collaborate more with externals than internally	2	Q4 aggregated on time Pattern Search Attribute = internal/external researcher	Find author. Pattern = an author who collaborates more with externals than internals
3	P1	Which people are more likely to have a journal publication with an external collaborator than with internal collaborators?	2	Q4 aggregated on time Pattern Search Attribute = internal/external researcher (node); publication type (edge)	Find author. Pattern = an author who collaborates more with externals than internals on journal publications
4	P1	Which are the people that sit between groupings and join groups together?	3	Q4 aggregated on time Pattern Search Structure	Find author. Pattern = bridge/hub nodes
6	P1	Is there any group that is totally unconnected to the rest of the school?	3	Q4 aggregated on time Pattern Search Structure	Find author group. Pattern = disconnected component.
7	P1	What is the strength of connection between each of the research centres?	2	Elementary structural comparison (aggregated on time) Plus Q1 Inverse Lookup Attribute = research centre affiliation	Structural comparison (between subgroups) to find strength of connection. Q1 inverse lookup to find authors associated with each research centre.
8	P1	Is anyone in the wrong research centre (going by their paper collaborations)?	2	Q4 aggregated on time Inverse lookup Attribute = research centre affiliation	Find author. Pattern = authors who collaborate more often with authors from outside their research centre
9	P1	In what ways have people shifted their collaborators over time?	3	Q4i Behaviour Characterisation Structure	Change in the structure of the network over time

		Participant task	Rating (1-4)	Framework category	Notes
10	P2	How do individuals/centres rank in terms of productivity?	2	Q4 aggregated on time Behaviour Characterisation Attribute only (ranking pattern) Attribute = publication count Plus Q1 Inverse Lookup Attribute = research centre affiliation	Pattern reported is a ranking pattern, where individuals/centres are ranked in terms of their publication count. Q1 inverse lookup to find authors associated with each research centre.
11	P2	How do individuals/centres rank in terms of levels of collaboration?	2	Q4 aggregated on time Behaviour Characterisation Attribute only (ranking pattern) Attribute = some measure of collaboration e.g. (ratio of) single/co-authored publications Plus Q1 Inverse Lookup Attribute = research centre affiliation	Pattern reported is a ranking pattern, where individuals/centres are ranked in terms of a measure of collaboration. Q1 inverse lookup to find authors associated with each research centre.
12	P2	At what point in their time within IIDI do individuals start producing collaborative work with others?	2	Q4i Pattern Search Structure	Find time. Pattern = appearance of co-authoring. (NB Search may best be carried out on the set of ego networks.)
13	P2	Do patterns of collaboration vary from research centre to research centre?	2	Q4 aggregated on time Direct Comparison Structure Plus Q1 Inverse Lookup Attribute = research centre affiliation	Comparison between structural patterns associated with research centres. Q1 inverse lookup to find authors associated with each research centre.
15	P2	Where there is little evidence of internal collaboration, are	2	Q4 aggregated on time Pattern Search Structure	Pattern search to first find authors who are not very collaborative

		Participant task	Rating (1-4)	Framework category	Notes
		these individuals non-collaborative, or are their collaborators elsewhere?		Plus Q4 aggregated on time Behaviour Characterisation Structure	(pattern = authors with a limited pattern of collaboration). Behaviour characterisation on the identified subgraph with regard to proportions of internal/external collaborators.
17	P3	Who is consistently a first author (does most of the work, active researcher)?	3	Q3 Pattern Search Attribute = publication count by author order	Find author. Pattern = authors who consistently have high levels of first authoring and lower levels of other positions of authoring. This can be handled in the same way as publication type e.g. finding people who mainly publish journals. We can either think of it as dealing with an attribute whose values are a set, or dealing with multiple attributes, where particular values/patterns are specified for each. "Consistently" implies that we are looking for a pattern over time i.e. where first authoring has a high value in all/most time periods.
18	P3	Who is consistently a last author (does least amount of work, supervision role only)?	2	Q3 Pattern Search Attribute = publication count by author order	As above (17), but pattern is that of a "consistent last author".
19	P3	Who is publishing most (speculative)?	2	Q4 aggregated on time Pattern Search Attribute only (ranking pattern) Attribute = publication count	Find authors. Pattern = top ranked publishers.
20	P3	Who is publishing only journal papers	3	Q3 aggregated on time Pattern Search	Find authors.

		Participant task	Rating (1-4)	Framework category	Notes
		(quality over quantity)?		Attribute = publication count by type	Pattern = high journal and no/low other types of publication. [NB similar to 17, but no notion of time included in this question]
21	P3	Who is collaborating without external partners?	2	Q4 aggregated on time Pattern Search Attribute = internal/external researchers	Find author. Pattern = authors whose ego networks have no external collaborators.
22	P3	Who is collaborating with external partners?	2	Q1 Relation Seeking Structure + attribute Attribute = internal/external	Although this appears to be a variation of 21, this is strictly speaking Q1 relation seeking (between values of attributes and at the same time, between references). We want to find authors that are connected but have different values of internal/external attribute i.e. relation between authors = *linking*; relation between values = *different* values of internal/external attribute.
23	P3	Who never collaborates?	2	Q4i Pattern Search Structure	Find authors. Pattern = isolates at all time points.
24	P3	Who always collaborates?	2	Q4i Pattern Search Structure	Find authors. Pattern = author who collaborates at all time points
25	P3	Who only collaborates with the same co-authors?	2	Q4i Pattern Search Structure	Find author. Pattern = ego network that does not change over time.
26	P3	Who has a mixed profile, name position varies dramatically, suggesting that they are almost always interested in	4	Q3 Aggregated on time Pattern Search Attribute = publication count by author order	See 17 (NB no mention of time in this question, hence Q3 aggregated on time). Find author.

		Participant task	Rating (1-4)	Framework category	Notes
		contributing whatever is needed?			Pattern (value) = a variety, or even distribution, of author order positions
27	P3	Who was active but now never publishes?	1	Q3 Pattern Search Attribute = publication count	Find author. Pattern = decreasing publication count over time.
28	P3	Who now only publishes book chapters? (winding down career)	1	Q3 Pattern Search Attribute = publication count by type	Find author. Pattern = increasing book chapter and decreasing other types of publications.
29	P4	How many times have 2 or more selected individuals published together?	3	*For a pair of authors:* Q3 aggregated on time Behaviour characterisation Attribute = publication count by type *For a group of authors:* Q4 aggregated on time Behaviour characterisation Attribute = publication count by type	NB pattern reported in terms of total instances of co-publishing
31	P4	Who is a new potential collaborator? Based on who they have published with previously	4	Q4 aggregated on time Behaviour characterisation Structure	Coded generally - the participant wants to understand the structure of the network in order to then make judgements about who potential collaborators might be.
32	P4	Who might I want to speak to for advice on writing an article? Based on their experience/number of publications/type of publication	3	Q3 Pattern Search Attribute = publication count by type	Find author. Pattern = one that suggests the author is experienced in writing articles e.g. increasing/high numbers of publications of a particular type over an extended period of time.
33	P4	Who is still currently research active? Based on recent publications	3	Q3 Pattern Search Attribute = publication count	Find author. Pattern = “currently research active” e.g. x level of publishing in recent years

		Participant task	Rating (1-4)	Framework category	Notes
34	P4	What types of articles are my colleagues publishing?	4	Q3 aggregated on time Behaviour characterisation Attribute = publication count by type	
35	P4	Who is still currently in the School?	2	Q2 Relation seeking Attribute = existence	Relation seeking involving the sets of authors that exist in the network at two different time points. Relation = authors that exist in the set of authors in both the current and previous year (set relation)
36	P4	What does the publication history of my colleagues look like?	4	Q3 Behaviour characterisation Attribute = publication count/type And/or Q4i Behaviour characterisation Structure	In this case we may want to look at a colleague's publication counts over time (Q3) and/or their pattern of co-authoring over time (Q4i – ego network)
38	P5	Given who I have co-authored with, who else am I likely to find as a good partner? (ie who is near me in the network)	3	Structural relation seeking (aggregated on time) Plus Q1 Inverse Lookup Attribute = author name	First find the author of interest ('me') using Q1 inverse lookup. Then find the co-authors' co-authors. Relation = connection at x distance, to the specified author.
39	P5	Who are the most productive publishers 'near' me in the network? Being able to filter by time period – eg 1-3 years – and publication type (journal). I'd ideally like to know who consistently reaches that magic 3* level, but that's not in this data set.	3	Structural relation seeking (aggregated on time) + additional constraint on node attribute Attribute = publication count by type Plus Q1 Inverse Lookup Attribute = author name	First find the author of interest ('me') using Q1 inverse lookup. Authors 'near me' = authors connected at x distance to a given author. We want to find authors who are connected (at x distance) to author y, and have a particular attribute value (high publication counts). This is structural relation

		Participant task	Rating (1-4)	Framework category	Notes
					seeking with an additional constraint on node attribute value. NB we perform this task on the network aggregated on time – either the whole time period or a subset of time.
40	P5	Who are the most experienced researchers 'near' me in the network? (ie who could I go to for advice)	3	Q4ii (time over graph) Pattern search Attribute = publication count by type Plus Q1 Inverse Lookup Attribute = author name	Q1 inverse lookup to find author of interest ('me') Pattern = experienced researchers (e.g. high levels of publications over an extended time period – see 32), connected to the author (at x distance).
41	P5	Who has just entered the network near me (and I need to find out more about)?	2	Q1 Inverse Lookup Attribute = author name Plus Structural relation seeking Plus Q2 Relation seeking Attribute = existence	Inverse lookup to find 'me'. Structural relation seeking to find authors connected at x distance to a given author in current year and in previous year. Relation seeking to find newly arrived authors (similar to 35). Relation = the set of authors that exist in the current year but not the previous year (set relation); performed on subgraph.
42	P6	Percentage of publications co-authored with externals, comparing research centres.	2	Q4 aggregated on time and graph Direct Comparison Attribute only Attribute = internal/external researchers	Inverse lookups to find authors belonging to research centres. Comparison is between subgroups (research centres), where an

		Participant task	Rating (1-4)	Framework category	Notes
				Plus Q1 Inverse Lookup Attribute = research centre affiliation	aggregate value (expressed as a percentage) is reported for each group.
43	P6	Years with the highest number of publications for each author, relative to joining the department. (Which career phase is most productive)	1	Q4ii (set of temporal trends) Pattern search Attribute = publication count	Find time period(s) (relative to start date). Pattern = periods of high publication counts within the set of trends
44	P6	Average number of authors on each publication for each research centre, compared to the percentage of single author publications, across the research centres. (Does this show differences in disciplines?)	1	Q4 aggregated on time and graph Direct comparison Attribute 1 = (average) author count per publication for each research centre Attribute 2 = Percentage of single author publications for each research centre	Comparison is between research centres on two different attributes (rather than comparison between the two different attributes.
45	P7	…the existing dataset would be of passing interest to me in relation to understanding the past research activity of members in my group (CID)	*1*	Q3 Behaviour characterisation Attribute = publication count/type And/or Q4i Behaviour characterisation Structure	In this case we may want to look at each author's publication counts over time (Q3) and/or their pattern of co-authoring over time (Q4i – ego network) – see 36.
46	P8	How has Person X published over the years?	3	Q3 Behaviour characterisation Attribute = publication count (by type) And/or Q4i Behaviour characterisation Structure	Q3 for publication counts over time; Q4i if we are interested in X's co-authoring patterns over time (see 36)

		Participant task	Rating (1-4)	Framework category	Notes
47	P8	How many co-authored papers are there between X and Y?	3	Q3 aggregated on time Dyad Behaviour characterisation Attribute = publication count	NB pattern reported in terms of total instances of co-publishing (as per 29)
48	P8	How many papers have been cross centre?	3	Q1 Relation Seeking Structure + attribute Plus Q2 aggregated on graph Direct Lookup Attribute = publication count	Relation seeking (between values of attributes and at the same time, between references). Relation between authors = connection. Relation between values = different values of research centre affiliation. Direct lookup to find number of publications.
49	P8	How many papers of a particular type were published in year X or between year X or year Y?	3	*In year x:* Q2 aggregated on graph Behaviour characterisation Attribute = publication count by type *Between year X or year Y:* Q4 aggregated on time and graph Behaviour characterisation Attribute = publication count by type	When reporting a single year, this is a Q2 task; for a time period, this is Q4.
50	P8	Who has published most? – over different time periods	3	Q4 aggregated on time (whole time or time period) Pattern Search Attribute only (ranking) Attribute = publication count	Find author. Pattern = top author.
51	P8	a. What's the average publication rate? b. Compared across individuals	3	a. Q4 aggregated on time and graph Behaviour characterisation Attribute = publication count b. Q3 aggregated on time Direct comparison Attribute = publication count	a. Lookup task to find the overall (i.e. all authors, all times) average publication rate. b. Comparison is either between the average for individuals, or between an individual's

		Participant task	Rating (1-4)	Framework category	Notes
					average and the overall average i.e. a specified value.
52	P8	When was the first paper published by X	3	Q3 Behaviour characterisation Attribute = publication count	NB partial pattern to be reported (start date only)
53	P8	When was the last paper published by X? or by X and Y together, or by team of X,Y, and Z.	3	*By X or by X and Y together:* Q3 Behaviour characterisation Attribute = publication count *By* X,Y, and Z: Q4 aggregated on graph Behaviour characterisation Attribute = publication count	NB partial pattern to be reported (end date only) For individuals and dyads, this is Q3; for groups this is Q4, with the subgraph treated as a single reference (i.e. aggregated on graph)
54	P8	Find any gaps in publication history for an individual or team	3	*For an individual:* Q3 Pattern search Attribute = publication count *For a team:* Q4 aggregated on graph Pattern search Attribute = publication count	Find time. Pattern = time period with no publications. As above (53), for individuals this is Q3; for groups this is Q4, with the subgraph treated as a single reference (i.e. aggregated on graph)
55a	P8	Who does X and Y publish with?	3	Structural relation seeking Aggregated on time	This involves two relations that need to be satisfied – i.e. find author(s) who publish with x *and* with y
55b	P8	How often are X and Y in the same team?	3	Q3 Behaviour characterisation Structure	Assume this is a question about the amount and frequency of co-authoring between X and Y (rather than total number of co-publications)
56	P8	Questions as above concerning a range of years, e.g. 2009-2015 (NB questions are 55a Who does X and Y publish with? And	3	a. Structural relation seeking Aggregated on time b. Q3 Behaviour characterisation Structure	As per 55a and 55b, but over a subset of years.

		Participant task	Rating (1-4)	Framework category	Notes
		55b How often are X and Y in the same team?)			
57	P9	Considering only one researcher e.g. JK how many of her publications are with the same group of researchers?	3	Q4i Pattern search Structure Plus Q4 aggregated on time and graph Behaviour characterisation Attribute = publication count	Q4i pattern search to find the set of authors who repeatedly publish with the ego (pattern = a set of authors who repeatedly publish with the ego). Behaviour characterisation to find co-publication counts for the ego network and report as total (i.e. aggregated on time and graph). Note that the lookup task is performed on relations.
58	P9	How many cross centre/disciplinary publications are there?	3	Q1 Relation Seeking Structure + attribute Attribute = research centre affiliation Plus Q2 aggregated on graph Behaviour characterisation Attribute = publication count	Find cross-centre publications using relation seeking (between values of attributes and at the same time, between references). Relation between authors = *linking*; Relation between values = *different* values of research centre affiliation. Once cross centre relations have been found, use lookup on linking relations to find the number of publications, and report as the aggregated total for all relations (i.e. aggregated on graph).
59	P9	The researcher's publications by year -	3	Q3 Behaviour characterisation Attribute = publication count (by type) And/or	Q3 attribute to look at publication counts over time; Q4i structure to look at co-authoring behaviour over time.

		Participant task	Rating (1-4)	Framework category	Notes
				Q4i Behaviour characterisation Structure	
60	P9	Type of publication by researcher -	3	Q3 aggregated on time Behaviour characterisation Attribute = publication count by type	Reported as total count of each type of publication (i.e. aggregated on time)
61	P9	Has the researcher collaborated with externals – if so can we have the details	3	Structural relation seeking Structure + attribute Attribute = internal/external plus Q1 Direct lookup Attribute = author/collaboration details	Relation seeking with an additional constraint on the node attribute value (external), plus Q1 direct lookup to find the details of the collaboration (publications etc.)/names of collaborators)
62	P10	How much is X publishing?	2	Q3 aggregated on time (subset) Behaviour characterisation Attribute = publication count	Aggregated on subset of time - assuming we want to know about recent publishing (rather than aggregated over all times), but not necessarily only the current year.
63	P10	Who's doing the work? (who are the first authors? Although it doesn't seem to be in the data, I'm also interested in the position of the authors. Usually, first authors are RA or PhD students)	2	Q3 aggregated on time Pattern search Attribute = publication count by author order	Find author. Pattern: authors with high levels of first author position and lower levels of other author positions (see 26)
64	P10	Who's working with whom?	2	Q4 aggregated on time Behaviour characterisation Structure	
67	P10	How much collaboration is taking place between groups?	1	Structural comparison Plus Q1 Inverse lookup	Find authors belonging to each research centre using inverse lookup; use structural comparison to find how much collaboration is taking place.

		Participant task	Rating (1-4)	Framework category	Notes
				Attribute = research centre affiliation	
68	P10	What types of publications are produced by an individual/group	2	*An individual:* Q3 aggregated on time Behaviour characterisation Attribute = publication count by type *A group:* Q4 aggregated on time and graph Behaviour characterisation Attribute = publication count by type	
69	P11	How does my publication rate compare to others?	2	Q3 aggregated on time Direct comparison Attribute = publication count	
70	P11	How does the quality and quantity of my publications compare to the targets set by the University	4	Q3 aggregated on time Direct comparison Attribute = publication count	Comparison with a specified value. Note that quality of publications is not included in the data.

Excluded tasks:

		Task	Rating	Reason
5	P1	What is the ordering of people when the number of collaborators? (would be better if the external collaborators were known and so could be distinguished)	2	Doesn't make sense
14	P2	Do patterns of collaboration vary according to job status?	2	As above, but comparison is between structural patterns associated with authors of different job statuses. Note that job status is not included in the data.
16	P2	Is it possible to identify mentorship relationships in the data?	2	High level task

30	P4	How many times have 2 individuals published together for the first time?	3	Doesn’t make sense
37	P5	**High level questions:** • Who would I be able to help? • Who would be interested in me? • Who do I need to make friends with? ☺	Not rated as too generic	High level task
65	P10	What topic is X working on? (I didn’t see it in the data, but presumably the publication reference must be available in the database, or at least the title? If it’s not, feel free to discard this question)	4	Research topic does not appear in the data
66	P10	What is the evolution of research topics for an individual/group over time?	4	Research topic does not appear in the data
71	P11	Who else is publishing in journals that interest me	2	Journal details are not included in the data

Appendix F Original Study Part 2 - Instructions to Participants

Instructions

When developing a visualisation system, it is important to understand what questions a person who will use the system would like to be able to ask of the data. We would like to develop a visualisation system to help **better understand collaborative working practices and publishing rates in the School of Computing**. We therefore would like to find out what questions people using the visualisation system would like to ask of the data that we have available.

In the first part of the study, you were asked to list the questions that you might like to ask of the data relating to publishing rates and co-authoring behaviour within the School over the years. A reminder of the data that we have available to us is included in data.docx.

For this part of the study, we have provided a list of potential questions covering different aspects of the data that might be of interest to ask. Please rate each question on a scale of 0-4 in terms of how interesting they are to you, using the following scale:

0 = of no interest
1 = slightly interesting
2 = moderately interesting
3 = very interesting
4 = extremely interesting

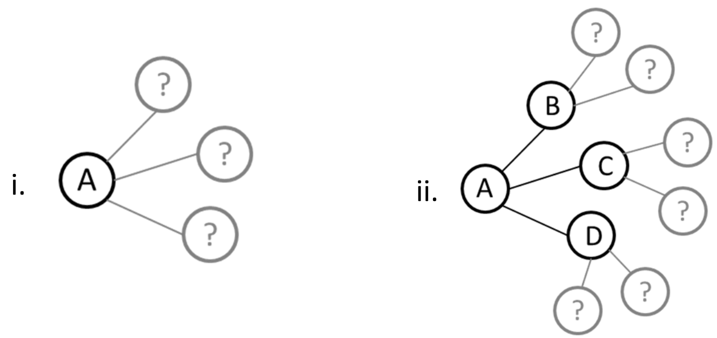
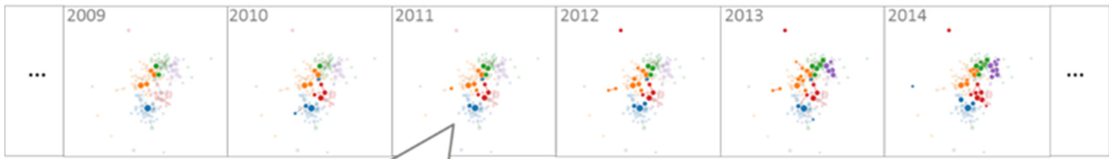
If you do not understand a question, please feel free to contact me for clarification (room C40; n.kerracher@napier.ac.uk; ext 2798). Otherwise, please simply note **DNU** (do not understand) in the relevant box.

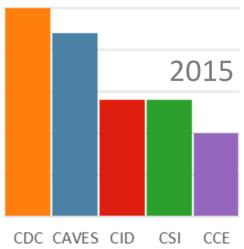
Please note that in the following questions:

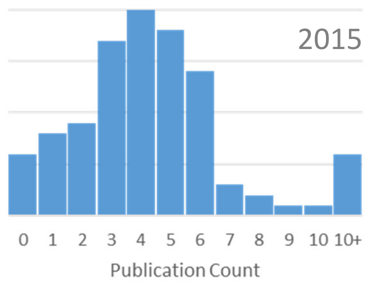
- “an attribute value” refers to the publication count or research centre affiliation associated with an individual author
- Where “Author A”, “Author B” etc. are used in the examples, it may be helpful to imagine an author that is of particular interest to you – for example, yourself, a colleague, a senior researcher etc.
- Images (charts, networks etc.) are used to help illustrate the question only and are constructed using synthetic data. There may be other, more appropriate ways to visualise the data when answering a particular question.

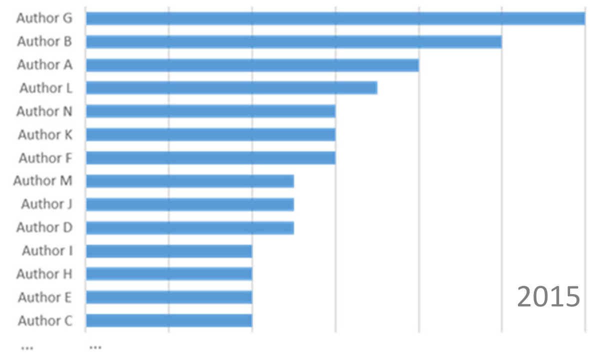
Questions

	Question	Your Rating
Q1 Direct Comparison (node)	Would it be interesting to compare attribute values between authors or between years? E.g. *compare Author A's publication count in 2015 and 2016; compare author A and author B's publication counts in 2015; compare author A's journal publication count in 2015 with their conference paper count.*	
Q1 Direct Comparison (edge)	Would it be interesting to compare co-authoring between pairs of co-authors or between years? E.g. *compare co-authoring between Author A and B in 2015 with that of Author B and C; Compare co-authoring between Author A and B in 2015 and 2016.*	
Q1 Inverse Lookup (edge)	Would it be interesting to find the authors associated with a particular amount of co-authoring (perhaps of a particular type of publication) in an individual year? e.g. *who are the authors who published 6 journal articles together in 2015?*	
Q1 Inverse Comparison (node, edge)	Say you spot some individual attribute values of interest e.g. particularly high publication counts. i. Would it be interesting to find and compare the authors associated with these attribute values or the years in which they occur? E.g. *Are the authors with the highest publication counts in 2014 and 2015 the same or different authors?; did author A have their highest number of publications before or after 2010?* ii. Would it be interesting to know if the authors associated with the attribute values are co-authors? E.g. *Did the authors with the highest publication counts in 2015 co-author?*	i. ii.
Structural Comparison	In a particular year, would it be interesting to know whether or not two specific authors were co-authors? E.g. *Did Author A and Author B co-author in 2015?*	
Structural Relation Seeking		i

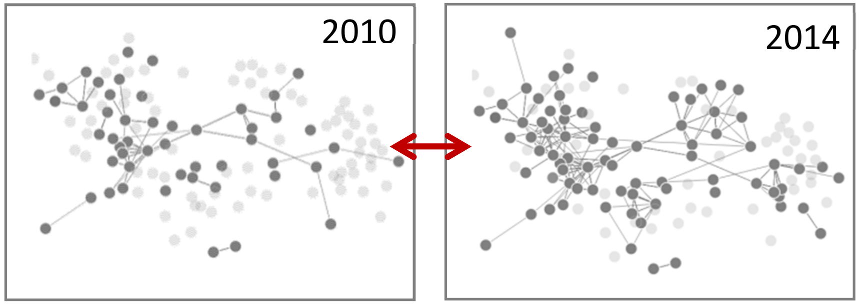
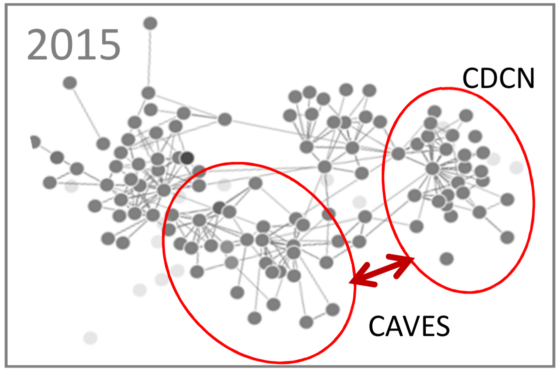
	 i. In an individual year, would it be interesting to find who a particular author's co-authors are? E.g. *who are Author A's co-authors?* ii. Would you like to know who an author's co-authors' co-authors are in a particular year? (i.e. those people who publish with a co-author, but not directly with the author) E.g. *who are Author A's co-authors' co-authors?* iii. More generally, would it be interesting to find pairs of co-authors? E.g. *who co-authors with whom?* Or pairs of co-authors with a certain level of co-publication e.g. *who has co-authored together at least X times?*	ii iii.
NB the following questions concern the network (its structure and associated attribute values) in an individual year:		
Q2 Behaviour Characterisation	  *This image shows a mock-up of the network in 2011 – circles represent authors, lines between them represent co-publication. Colour represents research centre affiliation, while size of*	

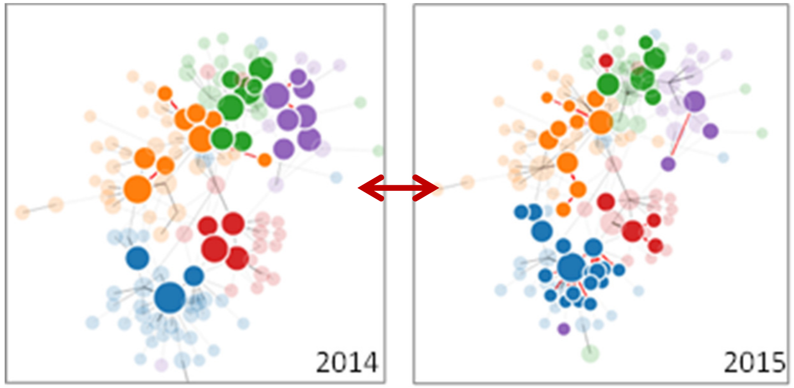
<table>
<tr><td></td><td colspan="2">Are you interested in understanding the network (or part of the network) in a particular year…</td></tr>
<tr><td>Q2 Behaviour Characterisation
- structure</td><td>…in terms of its structure? E.g. the size of the network, patterns of co-authorship, whether the network is tightly or sparsely connected (i.e. lots or little co-authorship), whether the network is completely connected or fragmented, whether there are groups of co-authors (clusters), whether there are authors who don't co-author (isolates), etc.</td><td></td></tr>
<tr><td>Q2 Behaviour Characterisation
- structure & attribute</td><td>…in terms of the relationship between the network's structure and attribute values? i.e. how attributes (publication counts and types, research centre affiliation) are distributed over the network. E.g. What does co-authoring and research centre affiliation look like in 2015? Do authors from the same research centre publish together or with authors from different research centres? What does co-authorship in a particular research centre look like? Do authors who have many co-authors in 2015 have higher publication counts?</td><td></td></tr>
<tr><td>Q2 Behaviour Characterisation
- attribute only</td><td>

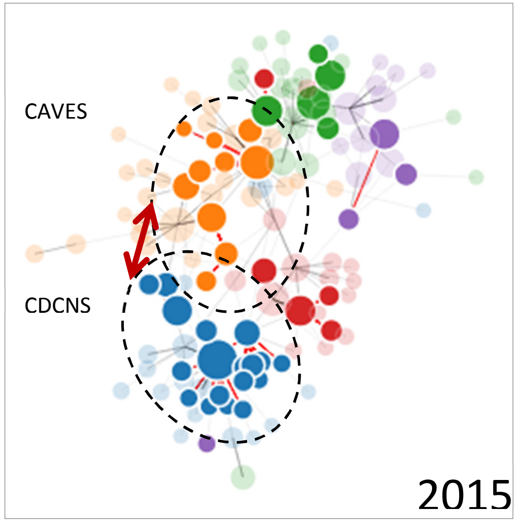




…in terms of attributes only? (i.e. without considering attributes in relation to network structure). For example, frequency distributions in a particular year (e.g. the number of authors in each research centre in 2015; the number of authors with 1, 2, 3, 4, 5, …, n publications) or ranking (e.g. of authors by number of publications/number of each type of publication).</td><td></td></tr>
<tr><td>Q2 Direct Comparison (structure)</td><td></td><td>i</td></tr>
</table>

	i  ii  Would it be interesting to compare… i. … the co-authoring network (or part of the network) at two different times? E.g. *compare the co-authoring behaviour and size of the network in 2010 and 2014* ii. …co-authoring behaviour in two different parts of the network? *E.g. compare co-authoring in CDCNS with co-authoring in CSI in 2015* iii. Would it be interesting to compare the network or part of the network to a specified pattern? *e.g. does the network in 2015 resemble a small world network?*	Ii iii
Q2 Direct Comparison (attribute &structure)	Would it be interesting to compare attribute distributions…	i

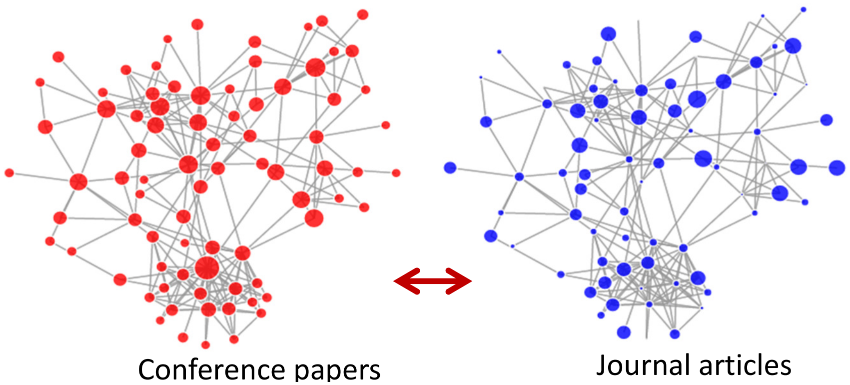




i. …over the network (or part of the network) at two different times? E.g. *compare the distribution of publication counts/research centres in 2014 and 2015*

ii. …in different parts of the network e.g. *compare the distribution of publication counts for CDNCS with CAVES in 2015*

iii. …with a specified pattern e.g. *how does the pattern in the network in 2015 compare with a pattern where authors with high numbers of co-authors also have high publication counts?*

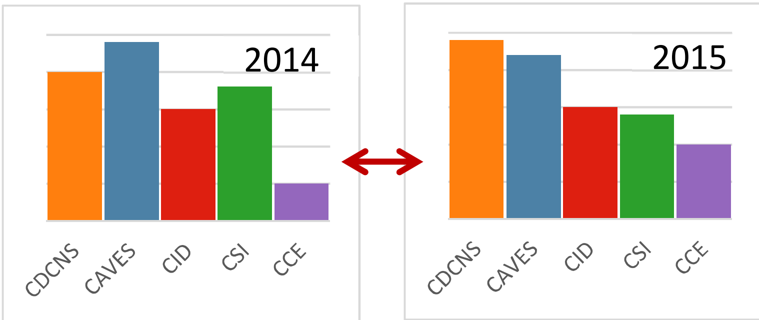


ii

iii

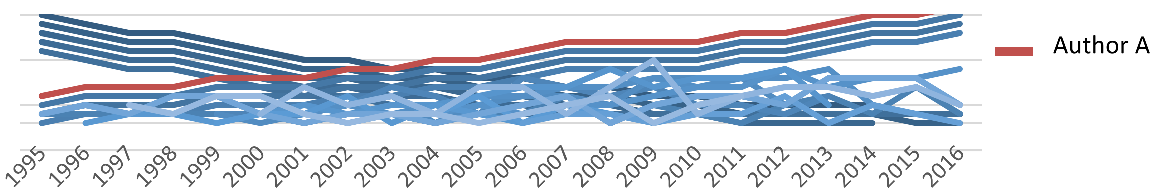
iv

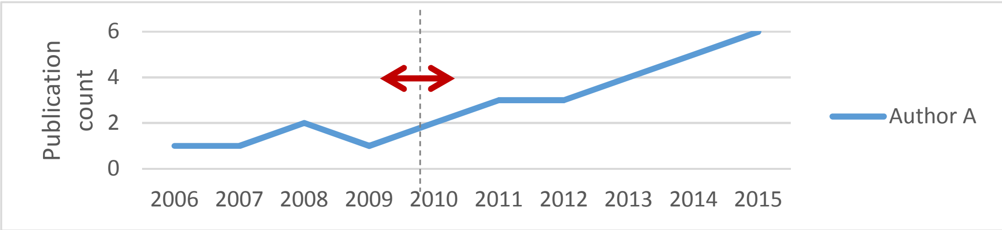
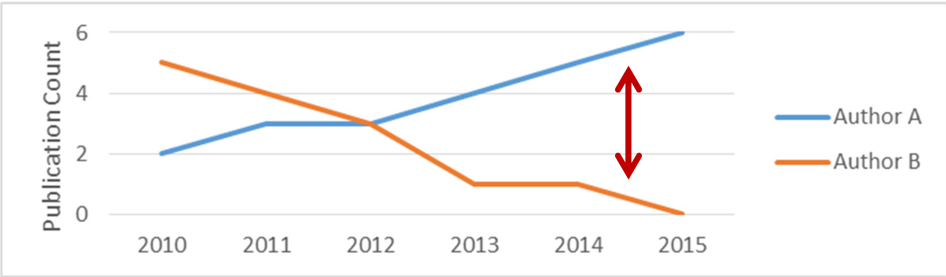
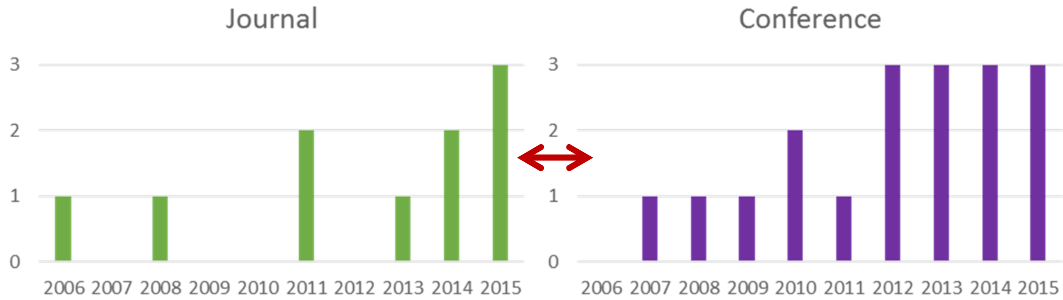
<table>
<tr>
<td></td>
<td>iv. Would it be interesting to compare the distributions of two different attributes over the graph (or part of the graph) e.g. compare the distributions of journal publication counts to that of conference paper counts in 2015?</td>
<td></td>
</tr>
<tr>
<td>Q2 Direct Comparison - Attribute Only</td>
<td>
Would it be interesting to compare frequency distributions or ranking patterns ….

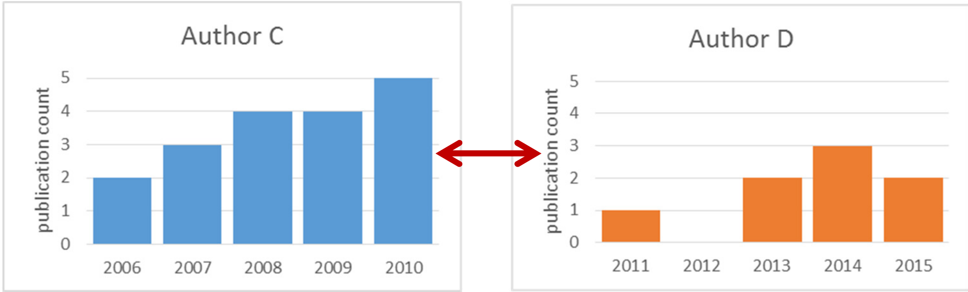


i. … between two years? E.g. compare the frequency distributions of authors in each research centre in 2010 with that of 2014; compare the rankings of authors by publication count in 2014 and 2015

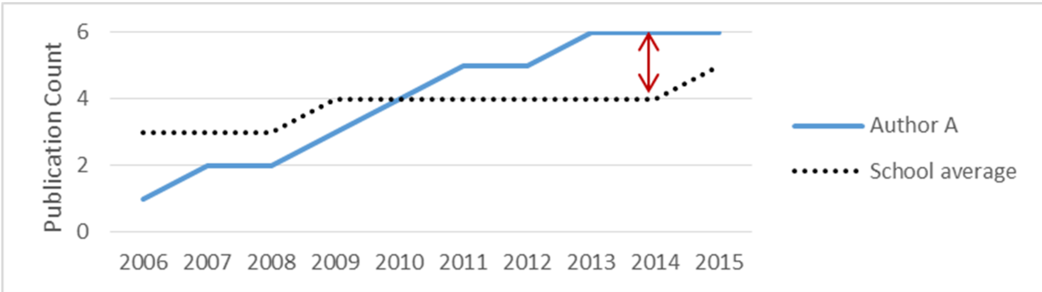
</td>
<td>i
ii
iii
iv</td>
</tr>
</table>

	ii. … for two different attributes? E.g. *compare the rankings of authors by journal publication count and conference paper count in 2015* *iii.* ...for different groups of authors? E.g*. compare the ranking patterns of publication counts for senior researchers and junior researchers* iv. …with a specific pattern? E.g. *compare the distribution of authors across research centres with a normal distribution*	
Q2 Inverse Comparison	Say you've found some patterns in the graph (at a particular time) that are of interest e.g. particular patterns of collaboration or attribute distributions over the graph. Would it be interesting to compare the authors and/or time periods associated with these patterns? e.g. *do the same or different authors belong to the tight clusters of co-authors seen in 2012 and 2013? Did the time at which very low publication counts are seen throughout the network occur before or after 2010?*	
Q2 Relation Seeking	Still thinking about patterns in the network in individual years (either the network's structure or attribute distributions over the network), would it be interesting to look for patterns in the network that are the same, opposite or different? e.g. *Are there any two consecutive years between which the network changes dramatically?* *Do any research groups exhibit* similar *patterns of co-authorship to that of CAVES? Are there any research groups with markedly* different *patterns of co-authorship? Do any research groups have* similar *distributions of publication counts?* *Are there any points in time when the network doesn't change in terms of its structure or attribute distribution? (i.e. remains the* same*)* *Are there any times when the distribution of journal publication counts is* similar *to that of the distribution of conference publication counts? Are there any times when the two distributions are very* different*?*	
NB The following questions consider trends over time for individual authors (e.g. trends in an individual's publication counts or changes in their research centre affiliation), and co-authoring between a pair of authors (e.g. the amount and frequency of co-authoring between two authors over time):		
Q3 Pattern Search (structure)	 Are there any particular temporal trends that you would find interesting? e.g. *for an individual author, are increasing or decreasing trends in publication of interest? Or patterns of movement between research centres? For a pair of authors, are there any patterns of co-authorship that you would find interesting e.g. continuous co-authoring or intermittent co-authoring?* If so, would you like to be able to browse or search the data and find the authors associated with these patterns and/or the time(s) at which they occur?	

Q3 Direct Comparison (attribute)	Would it be interesting to compare trends in attribute values over time associated with…  i. …an author at two different times *e.g. compare the trend in author A's publication count between 2006-10 and 20011-15?*  ii. …two different authors e.g. *compare the trends in author A and author B's publication counts between 2010 and 2015?*  iii. …two different attributes e.g. *compare the trend in author A's journal and conference paper counts over the whole time period.*	i. ii iii

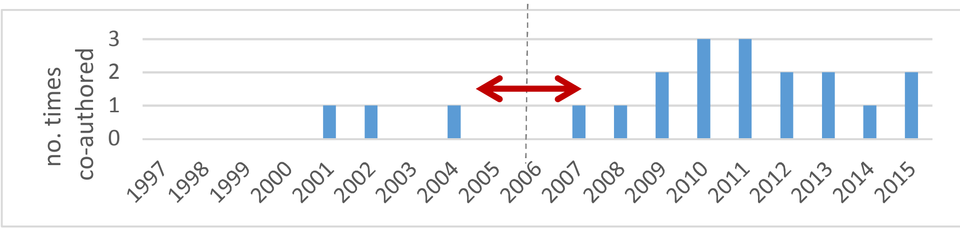
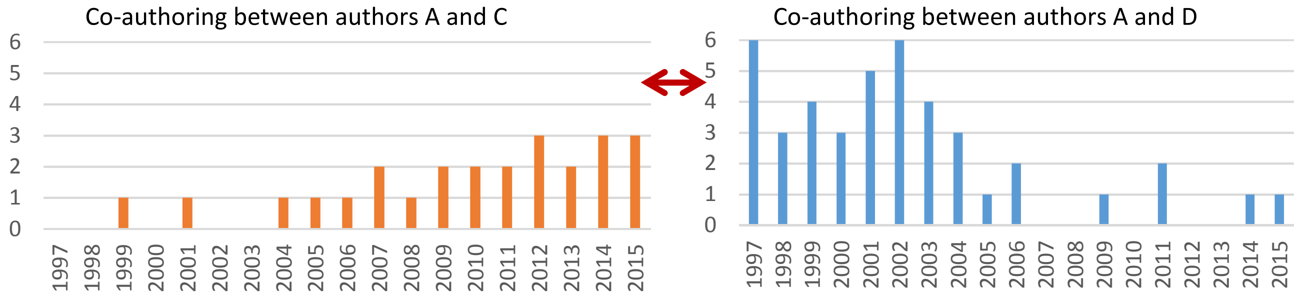
<table>
<tr>
<td></td>
<td>


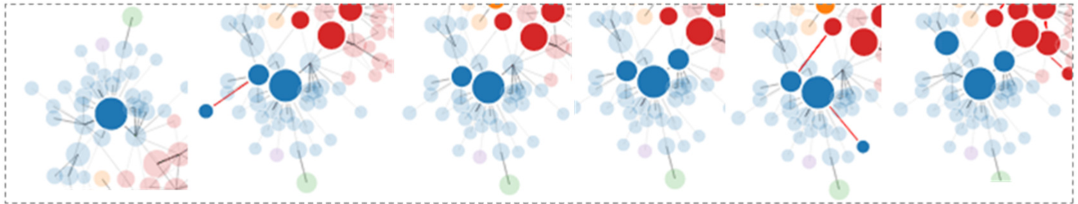
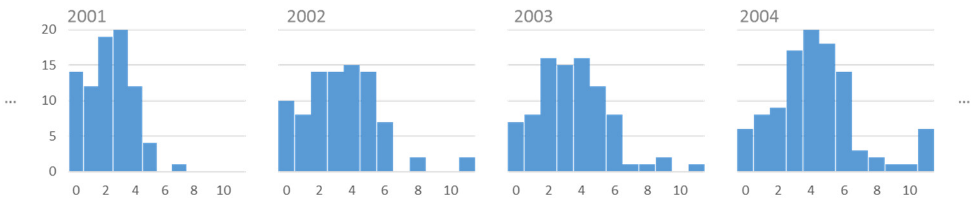
iv. ...some combination of the above, e.g. two different authors over two different time periods e.g. comparing the trends in publication counts of Author A and Author B in their first five years within the department.

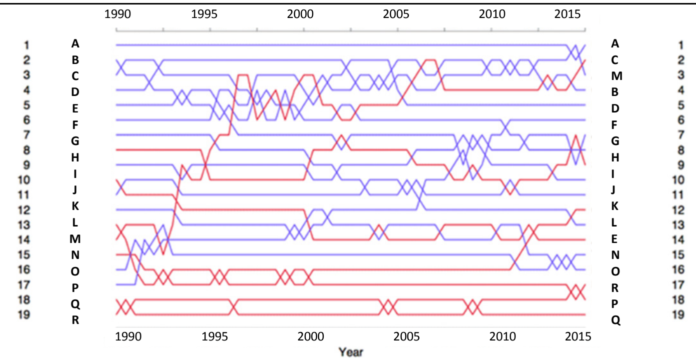


v. ...a specific temporal trend e.g. compare the trend in Author A’s publication count over time with the school average
</td>
<td>iv

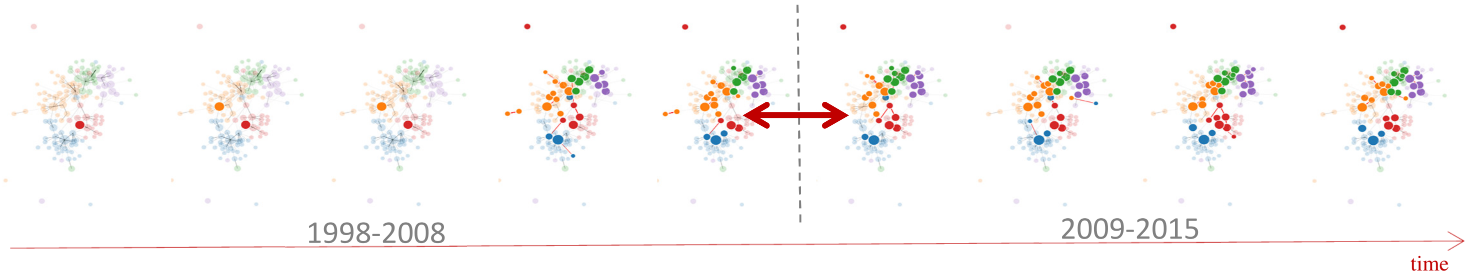
v</td>
</tr>
<tr>
<td>Q3 Direct Comparison (structure)</td>
<td>Would it be interesting to compare patterns of co-authorship between pairs of authors over time associated with...</td>
<td>i</td>
</tr>
</table>

	 *i.* ...a pair of authors over two different time periods e.g. *comparing the co-authoring behaviour of authors A and B before and after 2006*  *ii.* ...two different pairs of authors e.g. *comparing the co-authoring behaviour of authors A and C and A and D over the whole time period.* iii. ...a specific pattern of co-authorship e.g. *the average level of co-authoring in each year*	ii iii
Q3 Inverse Comparison	Say you've found some temporal trends of interest e.g. increasing publication counts or strong co-authorship patterns. Would it be interesting to compare the authors and/or time periods associated with these patterns? e.g. *is the author with very high rates of journal publications during the last REF period the same or a different author to the one with the very high rates of journal publications in the current REF period?* *Did author A and B's co-authoring collaboration begin before or after 2008?*	
Q3 Relation Seeking (structure, attribute)	Still thinking about individual trends over time (either patterns of co-authorship between pairs of co-authors or individual trends in attribute values), would you be interested in finding trends that are the same, opposite or different? e.g. *Are there any authors with the* same *trend in publication count as author A?* *Are there any times at which author A had a* similar *trend in publication count to that of 2010-14* *Does anyone else have a* similar *pattern of moving research centres to that of author D?* *Are there any pairs of co-authors who have a* similar *trend in co-authorship over time to that of Authors A and B?*	

	Which authors have similar *patterns of co-authorship over time?* *Do any authors have* similar *trends in both journal publication count over time and conference paper publication count over time?* *Are there any times during which an authors' trends in journal and conference publications are markedly* different?	
NB the following questions concern the network (its structure and associated attribute values) over time		
Q4i Behaviour Characterisation (Attribute + structure)	  Are you interested in understanding how the whole co-authorship network (or part of the network) evolves over time in terms of the relationship between the network's structure and attribute values such as publication count or research centre affiliation? E.g. *How does the distribution of publication counts change as the network evolves? Do authors with many co-authors have consistently higher numbers of publications over time? What about authors who continuously publish within the same co-author groups – is there a pattern to their amounts or types of publication? Is co-authorship between research centres changing over time?*	
Q4i Behaviour Characterisation (attribute only)	Are you interested in understanding how the whole co-authorship network (or part of the network) evolves over time in terms of attributes only (i.e. without considering attributes in relation to network structure), e.g.  Changes in frequency distributions - *How do frequency distributions (the number of authors in each research centre; the number of authors with 1, 2, 3, 4, 5, …, n publications) change over time?*	

	 Changes in rankings - *how do rankings of authors by number of publications/number of each type of publication change over time?*[53]	
Q4i Pattern Search (attribute & structure)	Are there any patterns in the changing network (or part of the network) over time that you would find particularly interesting, in terms of attribute and structure? e.g. *particular patterns of changing attribute distributions over the network such as rapidly increasing publication counts for better connected authors, while decreasing counts for less well connected authors, or changes in distribution of research centre affiliation over the network (which could signal a shift in collaborations between research centres).* Having spotted these patterns in the data, would you like to be able to find the time periods over which these patterns occur and/or find out who the set of authors associated with the changes are?	
Q4i Pattern Search (attribute only)	Are there any patterns in the attributes associated with the network over time (such as changing frequency distributions or rankings) that you would find particularly interesting? e.g. *particular patterns in frequency distributions over time such as shifting distributions of research centre affiliation; volatile or fixed patterns in author rankings by publication count.* Having spotted these patterns in the data, would you like to be able to find the time periods over which these patterns occur and/or find out who the set of authors associated with the changes are?	
Q4i Direct Comparison (attribute & structure)	Would it be interesting to compare the evolution of attribute distributions…	i.

[53] Bump chart showing changes in rankings over time – adapted from http://datatodisplay.com/blog/chart-design/communicating-changes-rank-time/



i. ...over the network (or part of the network) over two different time periods? E.g. *how does the evolution in distribution of publication counts/research centres over the network between 1998-2008 compare with that of 2009-2015?*

ii. ...in different parts of the network E.g. *how do the changing distributions of publication counts for CDNCS compare with that of CAVES between 2007 and 2015?*

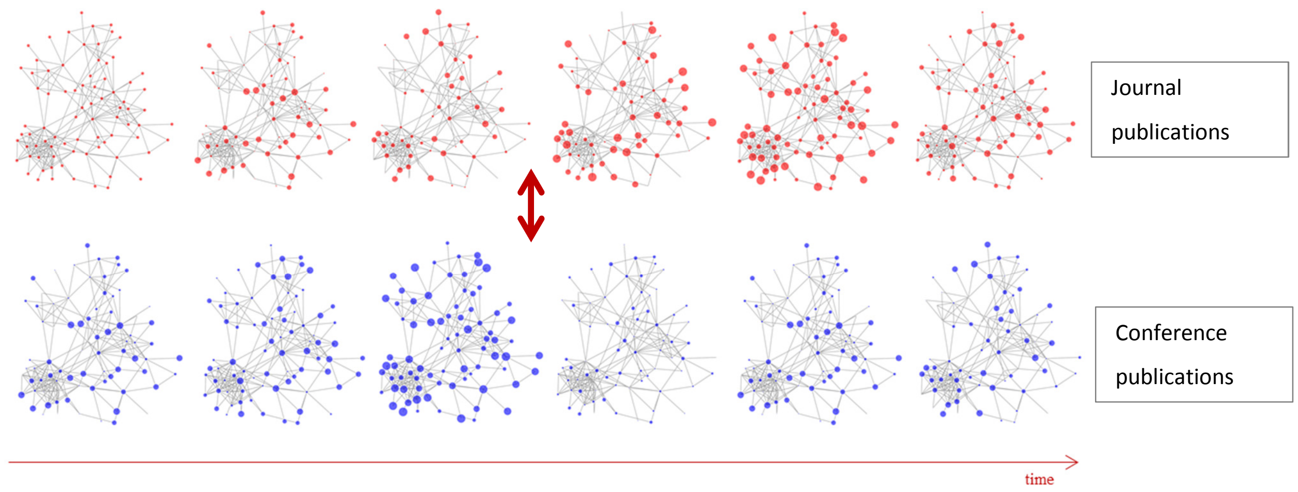
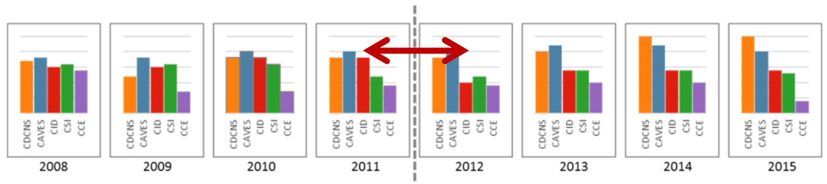
iii. ...with a specified pattern e.g. *increasing publication counts over the whole network*

iv. Would it be interesting to compare the changes in distributions of two different attributes over the graph (or part of the graph), over time? e.g. *how do the changing distributions of journal publication counts compare to that of conference paper counts?*

ii

iii

iv.

		
Q4i Direct Comparison (attribute only)	Would it be interesting to compare how frequency distributions or ranking patterns have changed over time….  i. … between two time periods? E.g. *how do the changing patterns of frequency distributions of authors in each research centre between 2008-11 compare with those of 2012-15?; how do the changing patterns of author rankings by publication count between 2000-2007 compare to those of 2008-2015*?	i

<table>
<tr>
<td></td>
<td>

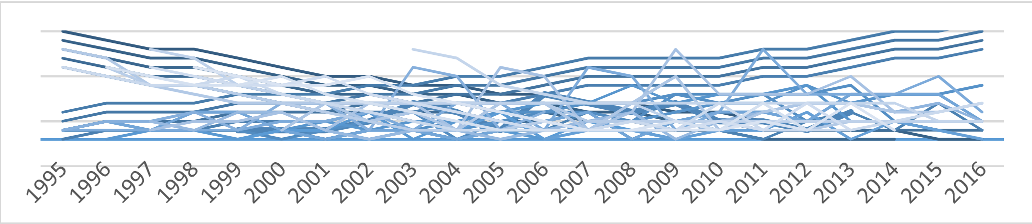

ii. … for two different attributes? E.g. how do the changing patterns of author rankings by journal publication count compare to changes in rankings of conference paper count over the time series?

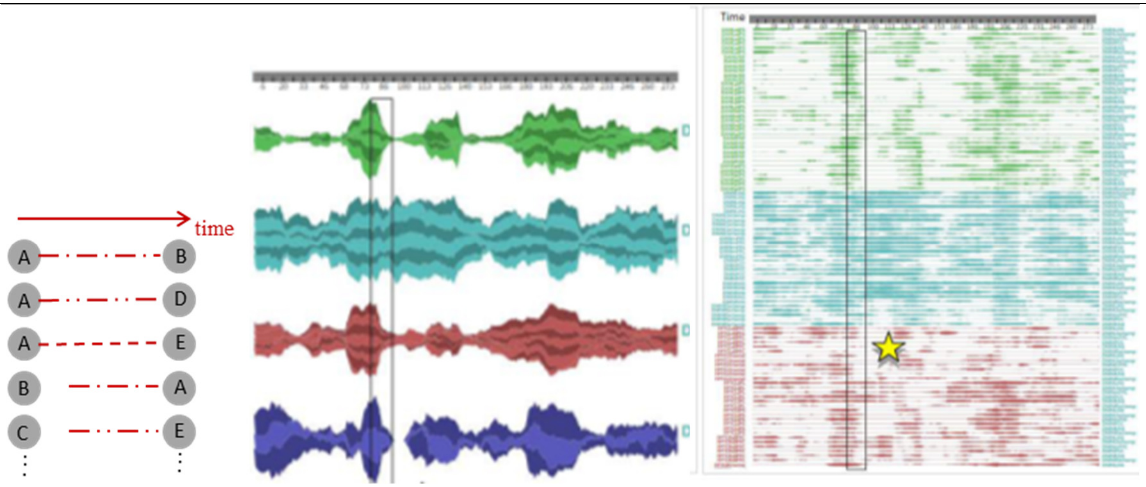
iii. ...for different groups of authors? E.g. how do the evolving ranking patterns of publication counts for senior researchers and junior researchers compare over the time series?

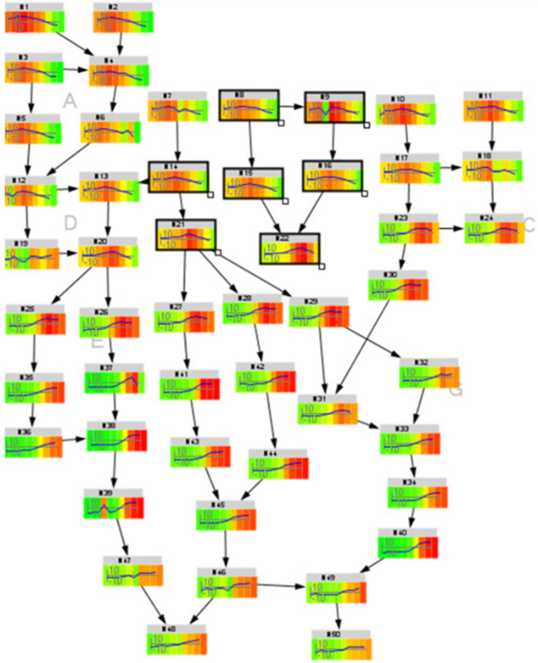
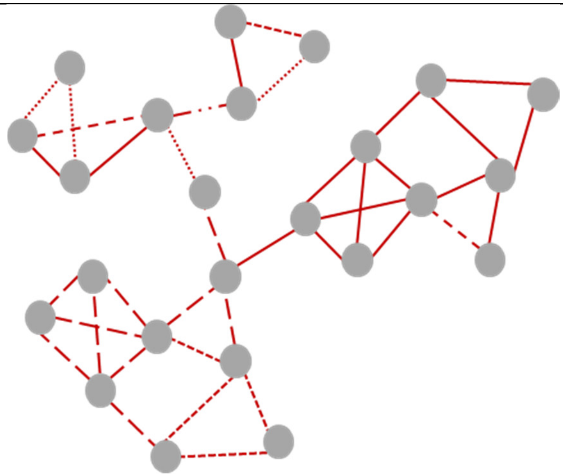
iv. …with a specified pattern? E.g. a pattern of stability in rankings/frequency distribution
</td>
<td>ii

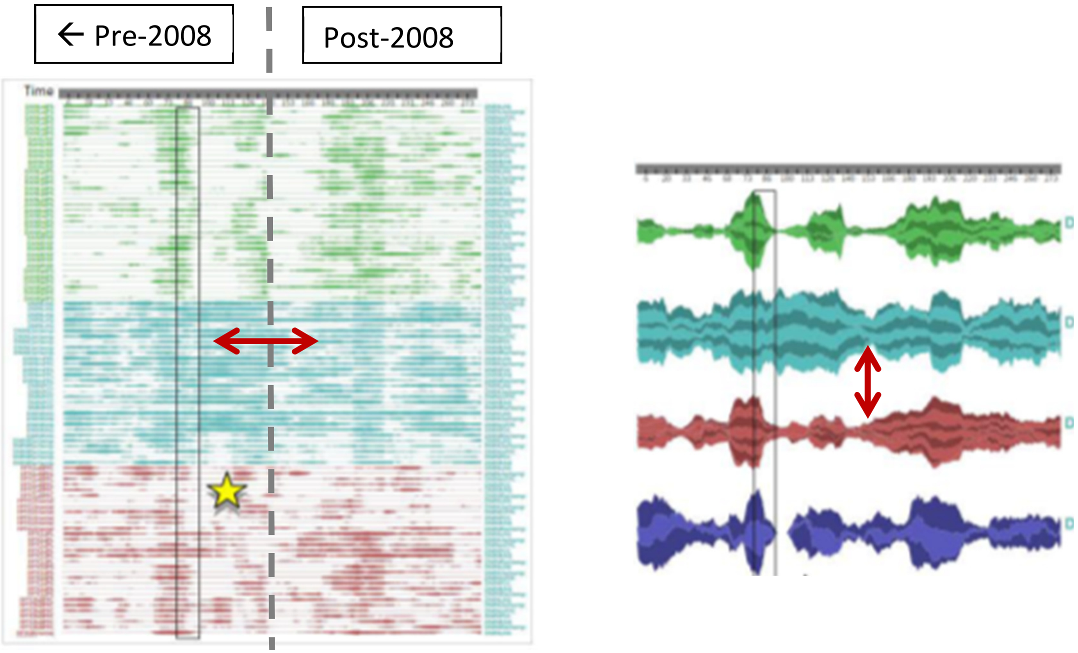
iii

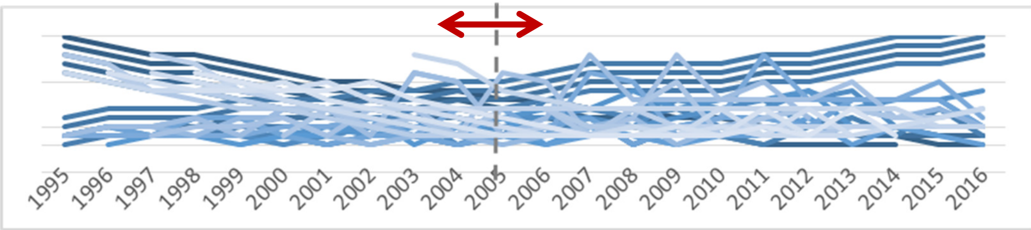
iv</td>
</tr>
<tr>
<td>Q4i Inverse Comparison</td>
<td>Still thinking about the changing network and attributes over time: say you’ve found some patterns that are of interest e.g. a period of increasing collaboration and publication count and a period of decreasing collaboration and publication count, or extreme changes in author rankings and a period of stability. Would it be interesting to compare the sets of authors and/or time periods associated with these patterns? e.g. are the same or a different set of authors associated with the increasing collaboration trend in 2004-2008 and the decreasing trend in 2010-15? Did the decreasing collaboration trend begin before or after a period of rapid increase in collaboration? Did the extreme changes in author ranking occur before or after 2010?</td>
<td></td>
</tr>
<tr>
<td>Q4i Relation Seeking</td>
<td>Again, still thinking about the network changing over time, would you be interested in finding evolving patterns in the network (either in co-authorship or attribute distribution) that are similar, opposite or different? E.g.
Are there any subgroups of authors with similar patterns in co-authoring over time? Is there a time period where the changing pattern of co-authorship is similar to that seen in 1996-2006?
Does any subgroup have a pattern of publication count distribution over the network, over time, similar to that of CSI? Is there a time period that has a changing publication count distribution similar to that of 2011-16?
Are there any time periods during which the distributions of journal publication counts and conference publication counts evolve in very similar or markedly different ways?</td>
<td></td>
</tr>
</table>

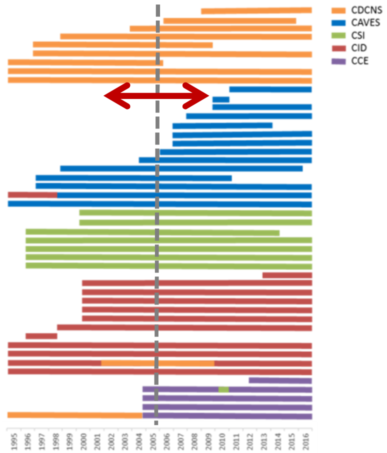
NB the following questions consider the set of individual authors' trends over time and the distribution of temporal trends over the network's structure.		
Q4ii Behaviour Characterisation (set of trends)	Thinking about trends over time, would it be interesting to explore the whole set of author trends, to see if there are any wider patterns within the School (or a particular research centre)…  Publication count over time (all authors)  Research centre affiliation over time (all authors) i …in attribute values over time e.g. *Are there general trends in publication amounts (e.g. peaks corresponding to REF dates or management changes)? Are there groups of authors whose publication counts are significantly increasing or decreasing over time? Are there wider patterns in staff joining and leaving the network, and/or research centre affiliations?*	i

	 ii …in the trends in co-authorship over time between all pairs of authors e.g. *whether the school is generally becoming more or less collaborative, whether there are particular time periods where co-authoring is low or high, or whether the patterns can be grouped into different categories (e.g.by type of collaboration - continuous co-authors, one-off co-authors, intermittent co-authors etc.)*	ii
Q4ii Behaviour Characterisation (distribution - attribute)	Would it be interesting to see how these individual temporal trends are distributed over the network…	

	 …in terms of attribute values e.g. *do authors closest to one another in the graph (i.e. collaborators) have similar trends in publication count?*	
Q4ii Behaviour Characterisation (distribution - structure)	 …in terms of co-authoring (represented in the diagram by links between authors) e.g. *are there clusters of similar temporal trends in co-authoring behaviours between pairs of authors over time?*	

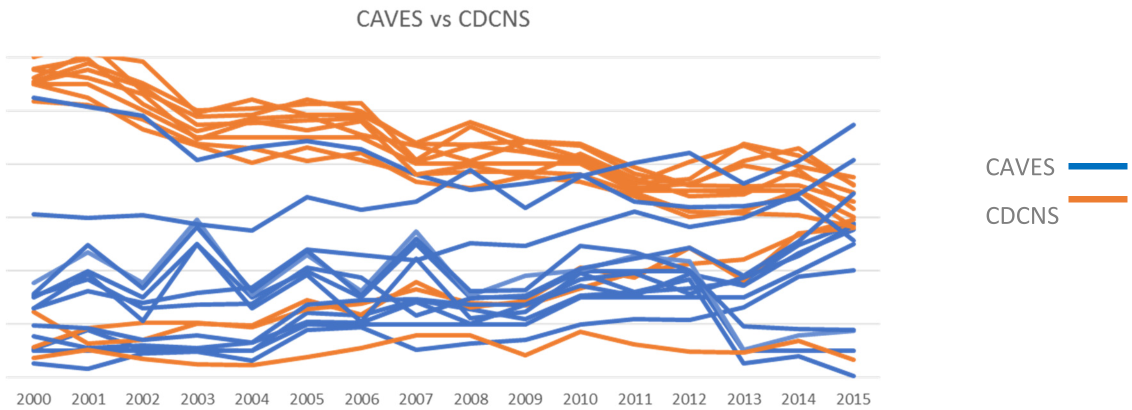
Q4ii Pattern Search (structure)	Would it be interesting to browse or search the data for specific patterns in the set of temporal trends in co-authoring (e.g. *clusters in the network where authors have similar temporal trends in co-authoring behaviours*), and find the authors associated with them and/or the time periods over which they occur?	
Q4ii Direct Comparison (structure, set of trends)	Would it be interesting to compare the wider patterns in the set of temporal trends in co-authoring?  i. ...between two different time periods e.g. *comparing the patterns in co-authorship pre- and post- 2008?* ii. ...between different groups of authors e.g. *comparing the wider co-authoring pattern for CDNCS with that of CAVES* iii. ...with a specified pattern e.g. *a general increase in co-authoring*	i ii iii
Q4ii Direct Comparison (attribute – set of trends)	Would it be interesting to compare wider trends in attribute values over time?	





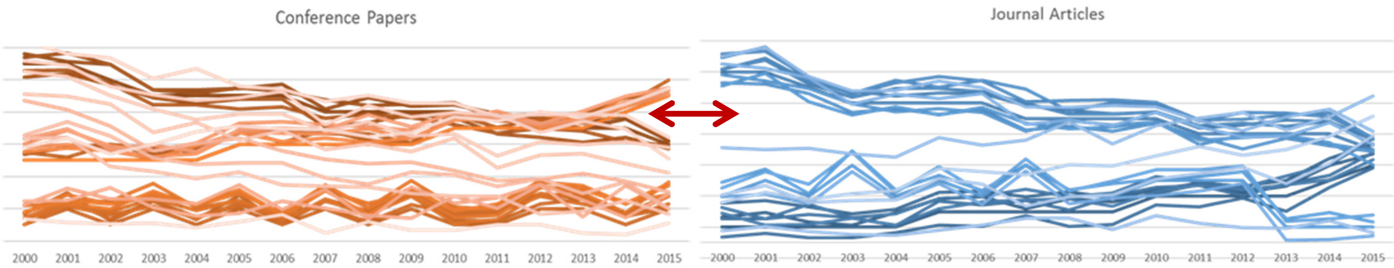
i.

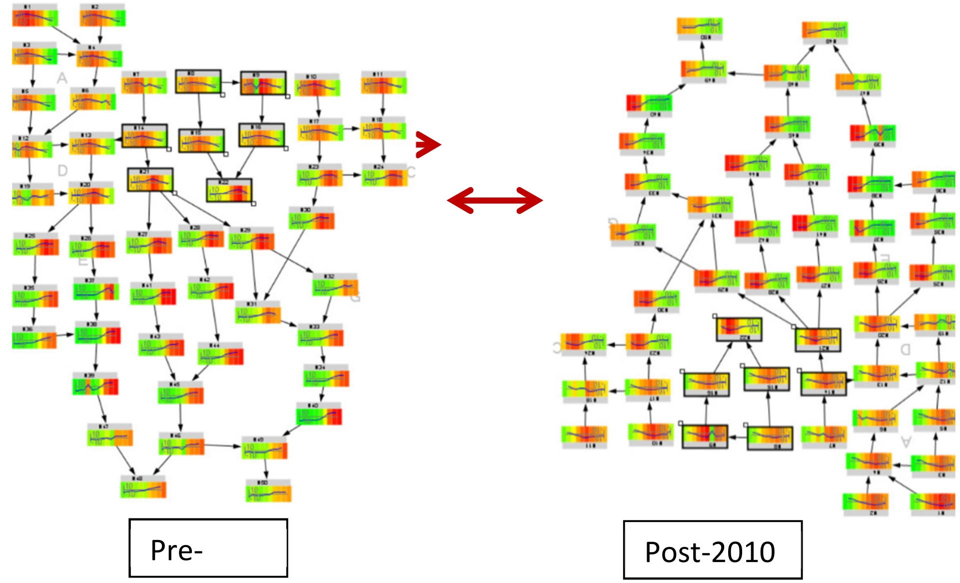
i. …between two different time periods? e.g. *how do the wider trends in publication counts pre-2005 and post-2005, or wider patterns in research centre affiliation/joining and leaving date, compare?*



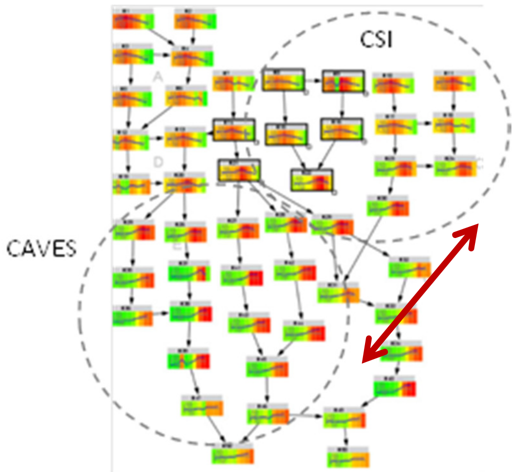
ii.

ii. …between two different sets of authors e.g. *how do the wider trends in publication counts over time for CAVES and CDCNS compare? Or the general trend in joining/leaving patterns between two centres?*

	 iii. …between two different attributes? E.g. *how do the wider trends in journal article publication counts compare with those of conference papers?* iv. …with a specific pattern? E.g. *a general increase in publication counts*	iii. iv.
Q4ii Direct Comparison (attribute – trends over network)	Would it be interesting to compare distributions of temporal trends in attribute values (publication counts, research centre affiliations) over the network?	i.



i. …between different time periods e.g. *how do the distributions of temporal trends in publication counts compare pre- and post-2010?*

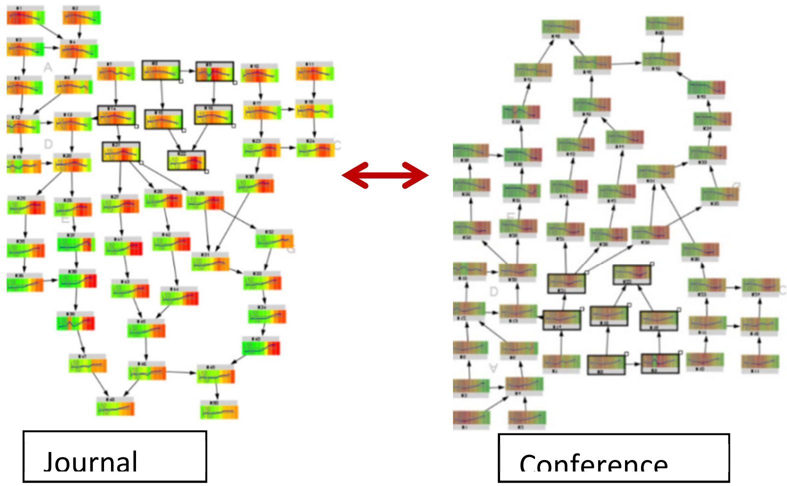
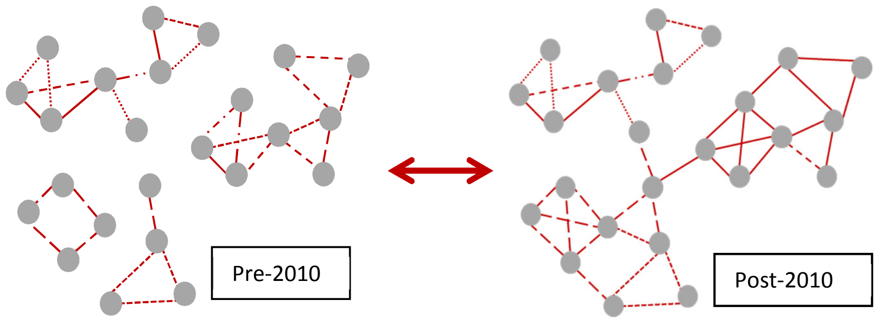


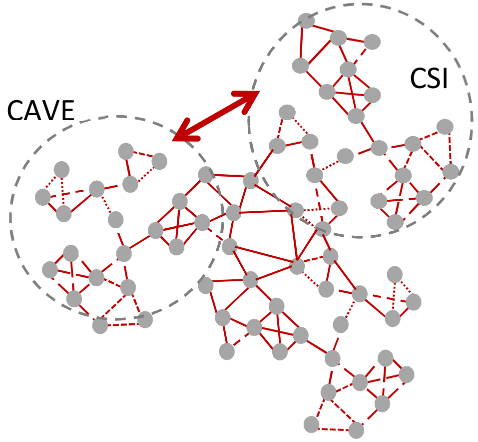
ii. …between different parts of the network e.g. *how do the distributions of temporal trends in publication counts compare for CAVES and CSI?*

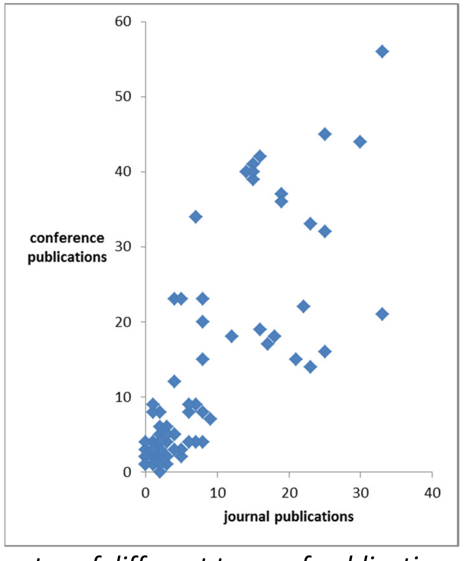
ii.

iii.

iv.

	 iii. ...between two different attributes e.g. *how do distributions over the network of temporal trends in journal publications and conference publications compare?* iv. ...with a specified pattern e.g. *uniformity in temporal trends across the network*	
Q4ii Direct Comparison (structure – trends over network)	Would it be interesting to compare distributions of temporal trends in co-authoring over the network (or part of the network)?  	i

	i. …between different time periods e.g. *how do the distributions of temporal trends in co-authoring compare pre- and post-2010* ii. …between different parts of the network e.g. *how do the distributions of temporal trends in co-authoring compare for CAVES and CSI between 2008-2016?* iii. …with a specified pattern e.g. *uniformity in temporal trends in co-authoring across the network* 	ii iii
Q4ii Inverse Comparison	Say you've found some wider patterns in the set of temporal trends/trends over the network that are of interest e.g. areas of the network with increasing trends in publication, a set of authors with falling publication counts, or an area of the network with similar patterns of co-authoring over time. Would it be interesting to compare the sets of authors and/or time periods associated with these patterns? e.g. *is the group of co-authors with increasing trends in journal publications the same or a different set of authors to the group of co-authors with increasing trends in conference papers? Did the general increasing trend in journal publication counts for authors in CAVES begin before or after 2010?*	
Q4ii Relation Seeking (set of trends)	Would you be interested in finding wider trends in the set of temporal trends that are similar, opposite or different? e.g. *are there any periods of time with similar global trends in publication count (e.g. periods of general increase or decrease in publishing); are there any periods of time with a global trend in co-authoring similar to that of recent years (2010-14)? Are there any research centres with a similar general trend in publication counts to that of CAVES?*	
Q4ii Relation Seeking (distributions)	Would you be interested in finding distributions of temporal trends over the network (either in co-authorship or attributes) that are similar, opposite or different? E.g. *Are there any co-author groups that have very similar distributions of temporal trends, but over different time periods?* *Are there any time periods during which the distribution of temporal trends is similar to that of 2010-16?*	

	Are there any time periods during which the distribution of temporal trends in journal publication counts is very similar (or markedly different) to that of conference paper counts?	
The following questions consider relationships such as influence/dependence and correlation between attributes, network structure, and structure and attributes		
Between attributes (Heterogeneous behaviours)	Would it be interesting to investigate the relationships between attributes? E.g.  *Is there a relationship between the publishing rates of different types of publication e.g. do people who publish many journal articles tend to publish fewer journal articles?* *Is there a relationship between research centre and publication count/type?* *Do high publication counts during one time period (e.g. a REF period) influence publication counts during later time periods?*	
Between structure and attributes	Would it be interesting to investigate the relationships between attribute values and network structure? E.g. *Is there a relationship between an author's position in the co-authoring network (e.g. central, on the periphery of the network etc.) and their publication count?* *Is there a relationship between an author's research centre affiliation and their position in the co-authoring network?* *Do certain patterns in the distribution of publication counts or research centre affiliation over the network precede particular changes in the networks' structure?* *Does the structure of the co-authoring network affect publication counts?*	
Between structures	Would it be interesting to investigate the relationship between the structure of the co-authoring network at different time points? Or whether changes in one part of the network affect other parts of the network? E.g.	

	Can we observe any mechanisms by which co-authoring relationships are formed? E.g. do authors with many co-authors increase their number of co-authors over time? (accumulative advantage); Do authors from the same research centre tend to co-author with one another? (homophily) *How does co-authoring at one point in time predict likelihood of co-authoring in future? Do authors seek to publish with new co-authors or maintain their already established relationships?* *Does the structure of the co-authoring network at one point in time affect the structure at later times?* *How do changes in one part of the network affect the rest of the network?*	

---

[i] Reduced from: ? g_2, λ: $\boldsymbol{f}(\mathbf{t}, \mathbf{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(\mathbf{t}, g_2) \in \mathbf{C''}$; $(\mathbf{t}, \mathbf{g_1})\ \lambda (\mathbf{t}, g_2)$

[ii] Reduced from: ? λ: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(\mathbf{t_2}, g_2) \in \mathbf{C''}$; $t_1\ \lambda\ t_2$

[iii] Reduced from ? t_2, λ: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, \mathbf{g}) \in \mathbf{C''}$; $\mathbf{t_1}\ \lambda\ t_2$

[iv] Reduced from ? t_2, λ: $\boldsymbol{f}(\mathbf{t_1}, \mathbf{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, \mathbf{g_2}) \in \mathbf{C''}$; $\mathbf{t_1}\ \lambda\ t_2$

[v] Reduced from ? t_2, g_2, λ: $\boldsymbol{f}(\boldsymbol{t_1}, \boldsymbol{g_1}) \in \mathbf{C'}$; $\boldsymbol{f}(t_2, g_2) \in \mathbf{C''}$; $(\mathbf{t_1}, \mathbf{g_1})\ \lambda (t_2, g_2)$

[vi] This is reduced from:? G', λ,: $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in G',\ x_2 = \mathbf{t}) \approx \mathbf{P_1}$; $\boldsymbol{\beta}(\boldsymbol{f}(x_1, x_2) \mid x_1 \in \mathbf{G''},\ x_2 = \mathbf{t}) \approx \mathbf{P_2}$; $(G', \mathbf{t})\ \lambda\ (\mathbf{G''}, \mathbf{t})$; i.e. all information (the graph subset, timepoint and pattern) is known in the second lookup subtask.