Achieve Semantic-based Precise Component Selection via an Ontology Model Interlinking Application Domain and MVICS

Chengpu Li, Xiaodong Liu and Jessie Kennedy
School of Computing
Edinburgh Napier University
Edinburgh, UK
{c.li, x.liu, j.kennedy} @napier.ac.uk

Abstract— **With the continuous increases in component varieties and size, it becomes more challenging to find components perfectly applicable for an application. A major obstacle for wider and smoother component reuse is the lack of automated and effective approaches to component specification and retrieval. This paper presents a domain-integrated ontology-based approach to holistic and adaptation-aware component specification and search. The work explores the possibility of combining the MVICS model with domain ontology, with financial domain as a case study. Such a combination enhances the function and application scope of the MVICS model by bringing more domain semantics into component specification and retrieval. The approach and its prototype tool support three distinctive features integrally, including Semantic-based Dynamic Component Retrieval, Flexible Interlink with Domain Model, and Result Component Profile. Furthermore, the inter-linkage between MVICS and domain models has resulted in improved retrieval precision and scope. The effectiveness of the approach and tool has been evidenced by user feedbacks via wide web use.**

Keywords-Automated component repository and retrieval; ontology-based component specification; domain ontology; linkage between ontologiese.

I. INTRODUCTION

At present Component-Based Development (CBD) fails to reach its full potential due to two reasons: firstly the lack of effective and automated methods for holistically and semantically specifying and retrieving existing components that precisely match user requirements [9]; secondly the specification model is either too domain specific to bear enough applicability or too generic to generate precise enough retrieval result. It is clear that approaches that are able to integrate the generic and domain-specific specification models in an easy and flexible way are highly needed. As we previously identified, the ontology in the existing approaches has too simple and/or monolithic structure and few relationships to deal with the specification and retrieval of modern components [5]. To resolve the above problem, an ontology-based approach is developed to achieve holistic and semantic-based component specification and then automatic and precise component retrieval. As a foundation, a Multiple-Viewed Interrelated Component Specification ontology model (MVICS) for component specification and repository was developed. Although the MVICS provides an ontology based architecture to specify components in a spectrum of perspectives, it is still fails to integrate knowledge of component based software engineering with application domain knowledge, and therefore is immature for real-life use.

In this paper, we focus on exploring the possibility of integrating the MVICS model with domain ontology models because any generic component specification model cannot practically survive without considering application domain. A financial domain related software system ontology model is established as a case study. Such integration enhances the function and application scope of the MVICS model by bringing more domain semantics into component specification and retrieval. The component retrieval method is now not only semantic-based but also dynamic and domain related by introducing the dynamic fiducial class weight for MVICS and the domain model. The result of retrieval includes the result component names, their accurate relevance rating, search information in both sub models of MVICS and the domain model and unsatisfied discrepancy, all are presented in a revised Result Component Profile. Another improved feature of the proposed approach is that the effect of possible component adaptation is also extended to specific domain related components, which enables a more systematic and holistic view in component specification and selection.

A prototype tool is then expanded to accommodate the above improvement on MVICS approach towards domain-integration. The component repository is built to accommodate components from financial domains. Extensive user feedbacks have been received based on case studies and a web-based version of the prototype. The new approach and tool has been evidenced to be more effective for the problem.

The reminder of the paper is organized as follows: Section 2 discusses related work with critical analysis. Section 3 introduces the Multiple-Viewed and Interrelated Component Specification ontology model briefly. Section 4 describes the financial domain related ontology and its linkage with MVICS. Section 5 describes holistic, dynamic and domain-related component retrieval. Section 6 describes the prototype tool and case study. Section 7 evaluates the approach and prototype based on practical use. Finally, section 8 presents the conclusion and future work.

II. Related Work

So far, the traditional component specification and retrieval approaches have been suffering from lower recall and precision [7][8][11][12], and are rather limited in accommodating semantics of user queries and domain knowledge. To solve these problems, ontology has been thus introduced to help understand the semantics of components, and domain model is used to capture application domain specific knowledge. In this section, we analyzed some typical work on component retrieval and repository with a focus on ontology-based domain knowledge related approaches, as follows:

When component-base software engineering (CBSE) became a research hot topic, the issue of providing specification of components through ontological models was noticed [4]. However, the research of ontology (information system) therein was not mature, and their ontological models were rather limited.

With the development of ontology technology, many approaches employed the ontology and domain model in the process of the component specification and retrieval. Sugumaran's approach [9] introduced domain ontology to apply the additional knowledge to augment or revise a user initial query. However, the domain ontology used in his approach is less sophisticated, and covers limited semantic information. Liu Quan's approach [6] presented a component description scheme with OWL language, and proposed an ontology based retrieval system architecture. Clearly Liu's scheme is not ontology based and its semantics are not computer-recognizable. In addition, he did not detail the information of domain ontology and the rules to link the ontology to user query. Braga [2][3] addressed the interoperability between component repositories, but only focused on the business components, leaving other kind of components (such as UI component, controller component and IT function component) undiscussed.

To summarize, although ontology-based domain model technologies have improved component specification and retrieval, the existing approaches still have the following limitations: i) ontology has simple and/or monolithic architecture and few relationships; ii) ontology models are either too domain specific to bear enough applicability or too generic to generate precise enough retrieval result; iii) lack of methods to link domain ontologies into generic component ontology.

III. Multiple-Viewed Interrelated Component Specification Ontology Model (MVCIS)

A holistic ontology model of component specification will provide the foundation for effective semantic matching in the component retrieval and improve the precision of component retrieval substantially. The MVICS ontology model has a pyramid architecture, which contains four facets: *function model, intrinsic model, context model* and *meta-relationship mode*. Each of the four models specifies one perspective of a component and as a whole they construct a complete spectrum of semantic-based component specification. All the four models are extracted from the analysis of CBSE knowledge and have extension slots for specific application domains. The first three models can be viewed as sub-ontology models, each of which describes one facet of component specification. The *meta-relationship model* is used to store four types of inter-relationships among the classes of the first three models. Meanwhile, OWL DL [1] is adopted to define the classes, individuals and relationships of the above models. A detailed description of the MVICS model is presented in [5].

IV. Integrating Domain Knowledge of Application Domains into MVICS

A financial domain related software system ontology was built for the purpose of a case study. This ontology mainly covers the basic financial operation software systems, and was built to collaborate with the MVICS to support the ontology-based financial domain component search. To obtain the semantic meanings between IT functions and financial domain operation, two mechanisms were established to link the financial domain ontology with the MVICS, namely Association Link (AssL) and Aggregation Link (AggL).

A. Financial Domain Related Ontology

Each class in this ontology represents one software system or module that carries out a financial operation. Superior-subordinate relationships have been used to describe the affiliations of the functions of these systems or modules. The top level classes include Asset Management Systems, Payments & Transfers Systems, Risk Management Systems and so forth. Their subordinates and the subsequent sub-subordinates and so forth constitute their sub classes and sub-sub-classes, till the most specific function units at the bottom level.

B. Ontology Linkage Method

Different from traditional ontology mediation, the connection between MVICS and financial domain related ontology neither uses the method of ontology merging to create a new ontology, nor uses the method of ontology mapping to make the same or similar ontologies to establish contacts. Our approach enhanced the function and application scope of the component retrieval by linking MVICS to different domain specification ontology with two mechanisms: Association Link (AssL) and Aggregation Link (AggL).

In our approach, for those classes in domain ontology which can be viewed as sub classes of a MVICS class, we call them "Association class". AssL is used to link these financial classes with their super class counterparts in MVICS. The Association classes in financial domain represent specific financial operations, which are viewed as a specialization of their MVICS super classes in financial domain.

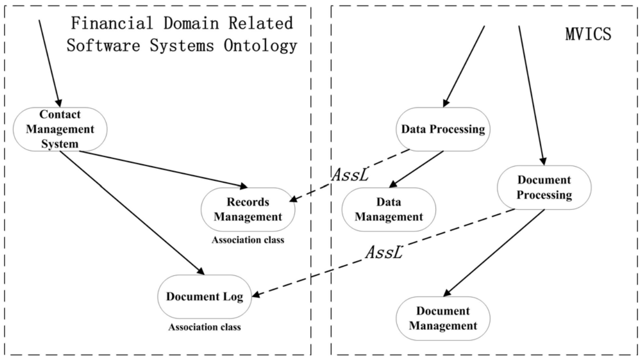

Figure 1. An example of Association

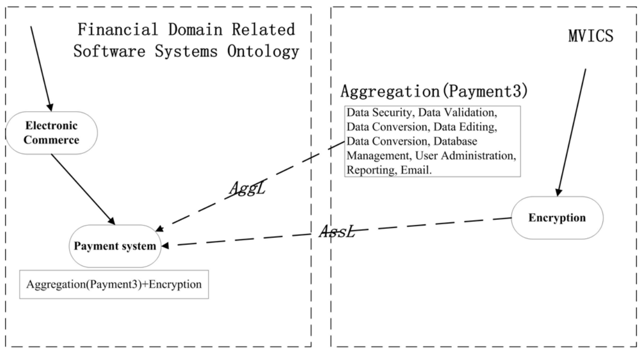

Figure 2. An example of Aggregation

Figure 1 shows an example of AssL. The class *Document Log* in financial domain ontology can be viewed as a sub-class of the class *Document Processing* in MVICS, because logging is a specific of document processing in financial sector, therefore the *Document log* link to *Document Processing* with AssL.

An Aggregation in MVICS is defined as a set of MVICS classes which work together to implement a larger function. Apart from the classes in the financial domain that can be linked in the way of AssL, other financial operation modules are more comprehensive and multifunctional. In this case, we first establish a set of reusable Aggregations in terms of the IT function in MVICS, to represent the function of financial domain operations. These reusable Aggregations are the function units of MVICS with minimum intersection of reusable MVICS functions. Each Aggregation is viewed as a reusable unit oriented to different financial operations. To link a domain model with MVICS, an AggL is defined as a link from a domain class to an Aggregation in MVICS. Classes in the financial domain ontology are linked to MVICS through AssL, where a counterpart super/subclass relation exists, and/or AggL, where a domain class is composed of one or a set of Aggregations in MVICS.

An example of AggL is shown in Figure 2. In total, class *Payment System* in financial domain ontology has the following ten functions which are already defined in MVICS, including *Data Security, Data Validation, Data Conversion, Data Editing, Data Conversion, Database Management, User Administration, Reporting, Email* and *system Encryption*. The first nine functions are composed into an Aggregation (Payment 3) in MVICS, and linked with AggL to class *Payment System.* In addition to the above nine functions, the class *Payment System* has an extra function *Encryption*, and this is expressed with an AssL from class *Payment System* to the MVICS class *Encryptio*n. Thus, the whole function of class *Payment System* is expressed by a combination of the AssL and AggL.

V. Holistic, Dynamic and Domain-Related Component Retrieval

"Holistic" here refers to that the MVICS is a comprehensive component specification model, and the approach considers a spectrum of respects in component specification and retrieval, such as adaptive component matching and result component profiles. In addition, the financial case study enhanced the function and application scope of the MVICS model and improved the Retrieval Method and Component Profile as follows.

A. Dynamic Class Weight Calculation

A method for basic class weight calculation is first proposed in [5], however our later findings show that the calculation has to be adjustable at run time because the constant evolution of components in function and QoS. Weight of class (W) is defined as the foundation for calculating the precision of result components. In each sub-model of MVICS, every class is given a weight to calculate the relevance of each search result. The weight assignment are formally defined as $W = (1+X)^n$, where n is the level of the layer in which the class locates, X = 0.5 for a class in the function model, X = 0.3 for one in intrinsic model, X = 0.2 for class in the context model [5].

The fiducial weights (X) of the classes in each model are given based on our experience. According to the testing data collected from the user, the fiducial weights will be updated dynamically after every 100 groups of user keywords are obtained. Each group of the keywords will be recorded and classified by the facets of the MVICS. The rules of dynamic fiducial class weight assignment are: the more frequently the keywords are used in a facet, the heavier fiducial weight of this facet is. Let N represent the occurring times of the keywords in a facet, the subscript f, i, c indicate the related facet (function, intrinsic or context). The weight assignment rules are defined as follows:

$$X_f = 0.5 \times \frac{N_f}{N_f + N_i + N_c} \qquad X_i = 0.3 \times \frac{N_i}{N_f + N_i + N_c} \qquad X_c = 0.2 \times \frac{N_c}{N_f + N_i + N_c} \quad (1)$$

Moreover, some users may wish only to be influenced by the views of the users in their own user group. Hence, the following rules are proposed to assign fiducial class weight for this purpose, i.e., in support of user group oriented component search, which can further improve the precision of search results. The superscript R, E, A indicate which user group is related, namely software engineering researchers, software engineers and amateurs.

$$X_f^{R,E,A} = 0.5 \times \frac{N_f^{R,E,A}}{N_f} \quad X_i^{R,E,A} = 0.3 \times \frac{N_i^{R,E,A}}{N_i} \quad X_c^{R,E,A} = 0.2 \times \frac{N_c^{R,E,A}}{N_c} \quad (2)$$

The weight of a search path (*Wp*) in the MVICS is the sum of the weight of the classes included in it.

The weight of financial domain ontology classes are given on the basis of MVICS class weights. As mentioned in section 4, AssL and AggL were developed to link the financial domain ontology to MVICS. For the class linked with a MVICS class via AssL, its weight is the same as the MVICS class it linked to. For a domain class linked with MVICS through AggL, its weight is the sum of weights of classes contained in the Aggregation. The *Wp* in the financial domain ontology is the weight of the matched class.

B. Domain Related Retrieval Algorithm and Precision Calculation

Based on linkage between MVICS and the financial domain ontology, the component search algorithm was refined. The financial domain keywords are identified as Function Keywords (FK). At the same time, the search tool records the information of the keywords, which will be used to refine the fiducial weights.

The prototype tool will then search the Function Keywords, together with Intrinsic Keywords (IK) and Context Keywords (CK) in the OWL files of MVICS and financial domain ontology. Meanwhile, it will record the search path of every keyword and then calculate the path weight by summing up every class weight in this path for MVICS matched class and by recording the class weight for financial domain ontology matched class. The search tool will record the components that link to the result class. After retrieval, a set of records is obtained for each keyword, which includes the result component name, the search path and its weight. The match precision of a result component (*Pc*) is calculated by the following dynamic fiducial weight, domain related formula, which is refined base on a previous unified formula proposed in [5]:

$$Pc = \frac{\sum_{r=1}^{a} WpFK_r}{\sum_{t=1}^{i} WpFK_t} \times X_f + \frac{\sum_{r=1}^{b} WpIK_r}{\sum_{t=1}^{j} WpIK_t} \times X_i + \frac{\sum_{r=1}^{d} WpCK_r}{\sum_{t=1}^{n} WpCK_t} \times X_c \quad (3)$$

VI. PROTOTYPE TOOL AND CASE STUDY

A prototype tool has then been expanded to accommodate the above improvement on MVICS approach towards domain-integration. A component repository is built to include components from financial domains. To evaluate both the approach and the prototype tool, a case study based on financial domain has been conducted with real-life scenarios of component search.

A. The Prototype Tool

The prototype tool has a simple user interface: A text area for the input of search keywords. The first column of option buttons lists the available domain ontology; the second column spreads out user-oriented options (i.e. Researcher, Engineer and Amateur). And a black panel for showing the summary of search results.

In this tool, the MVICS model, the domain ontology and the related Association classes and Aggregations files are implemented in OWL and located in corresponding OWL files. Figure 3 shows a system overview of the tool. User requirements are refined by the synonym operator first, the corrected and OWL defined keywords are mapping with the classes both in the MVICS ontology and domain related software system ontology. At the same time, the refined keywords are recorded by the requirement recorder. These records are used as primary data to update the fiducial weights of the MVICS classes by the following dynamic class weight assignment method.

On completion of the initial ontology-based searching, if the tool finds some matched classes and their instances (called "result components"), the precision calculator will further calculate the precision of each result component; if there is no matching class, the tool will then search for available adaptive methods or assets in adaptive component matching part. The proposed approach accommodates the impact of adaptation in the specification and selection of matching components. This

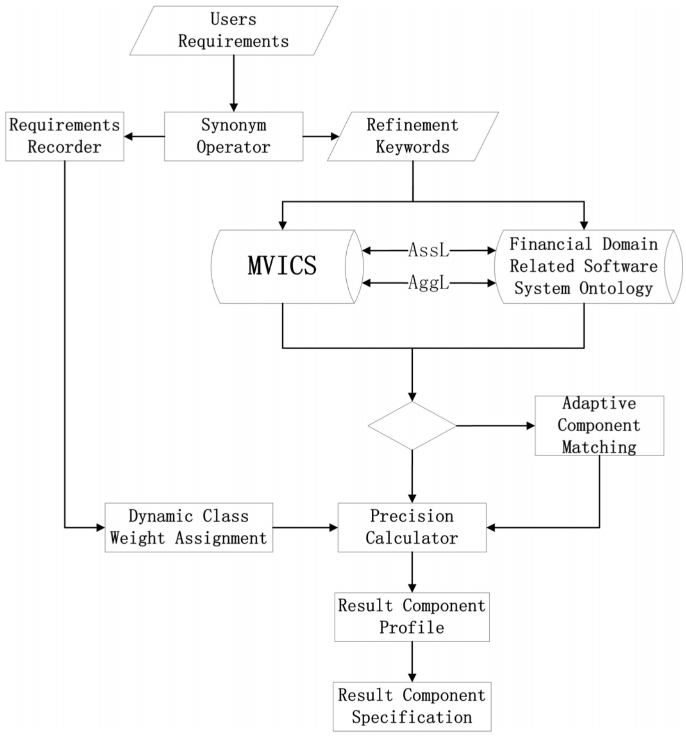

Figure 3. System overview of the prototype tool

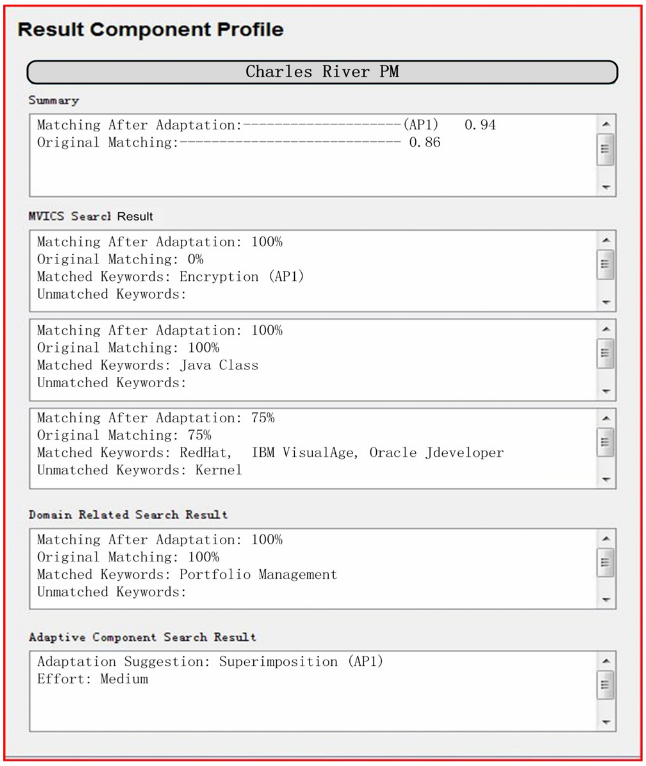

Figure 4. The Result Component Profile

unique feature will allow a more systematic and holistic view in component specification and selection. In MVICS and domain ontology, the adaptive components are linked to a class via an adaptation method or assets if the component becomes relevant to that class after adaptation with that method or asset. These methods or assets are defined as classes or instances in the MVICS and the financial domain ontology. After the adaptive component matching, the search results also go through the following precision calculator to compute the searching precision.

In contrast to most existing approaches, which present only the name and precision of the result component, our prototype tool provides a holistic Result Component Profile to help the user make the best decision in component selection. The revised profile includes the brand new domain related search result for user inspection.

B. Case Study

A retrieval scenario of developing an encrypted portfolio management system has been studied. We presume a user plans to search for relevant components with the following requirements: *Portfolio management, Encryption*, *Java Class*, *RedHat, Kernel*, *IBM VisualAge* and *Oracle Jdeveloper.*

Because the portfolio management system is a kind of financial domain related system, the user clicks the financial domain option button, and leaves user-oriented option buttons blank for original precision calculation. The search tool implements the keywords follows the process of prototype tool as mentioned in section VI. The search engine searched the keywords one by one in function, intrinsic and context sub-model of MVICS, and then financial domain ontology. Afterwards, the search engine search available adaptive methods or assets in adaptive component files.

The names of the result components and their precisions (calculate by the precision calculator) are displayed on the right show panel of the interface. When a result component is highlighted, its search result profile will pop up, as shown in Figure 4. The result component name “Charles River PM” is shown on the top button which can be further clicked to full structured textual specification. Below the component names, it is a search summary, including the original search precision (0.86) and the after adaptation search precision (AP1-Adaptation Path 1, precision is 0.94). The keywords matching information in MVICS model is listed in the middle of the profile. It comprises the keywords matching percentage in each sub model of the MVICS (original or after adaptation), the matched and unmatched keywords. As the shown example, the matching percentages in the three sub models of the MVICS are 100%, 100% and 75% after adaptation; the percentages are 0%, 0% and 75% originally. The keywords “Encryption”, “Java Class”, “RedHat”, “IBM VisuaAge”, “Oracle” and “Jdeveloper” are matched in the corresponding sub model respectively. The keyword “Kernel” fails to match in the context model. The following domain related search result display is new in the profile. It shows the search information of the domain related keywords which are collected with help of the linkage between the MVICS and the financial domain ontology. The only domain related keyword “Portfolio Management” is matched in the financial domain ontology model. Therefore, the percentages of original and after adaptation are all 100%, and on unmatched keywords. The available adaptation method(s) or asset(s) with their efforts are specified in the bottom adaptive component search result box.

VII. Experiments and Analysis

To verify the linkage between MVICS and domain ontology model, the prototype tool was transformed to a web application and published on the site http://ceres.napier.ac.uk/staff/chengpu/index.asp. This is a comprehensive project website, including project introduction, component specification, component publish and management, online tool test and questionnaires. Six hundred components were selected from component sale websites, with possible adaptation assets developed, and then were populated into a component repository. To compare with traditional component retrieval approach and other domain ontology-based approach, a SQL database search tool and a domain ontology-based component search tool were built, respectively. The domain ontology-based search tool uses the same financial domain ontology for refining the user requirements and specifying the component without MVICS. Software engineers, researchers and amateurs can use the applications and make comments via a questionnaire. To help the users to compare the function of

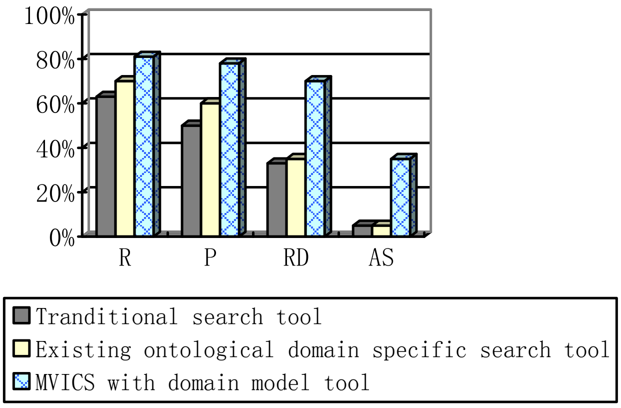

Figure 5. The level of satisfaction of the MVICS with domain model tool, existing ontological domain specific search tools and traditional search tools

different approaches, several financial domain related reuse scenarios with their matching components are provided. Users can test the search results by using given scenarios or producing their own.

The following procedures are suggested for comparison:

1) Using the provided scenarios and the corresponding result components (R_1), or proposing their own scenarios and realizing the suitable component (R_1) manually.

2) Using the SQL database search tool, existing ontological domain specific search tool and MVCIS plus domain model tool to search the same requirements and receive a set of search results (R_2), (R_3), (R_4) respectively.

3) Comparing R_2, R_3 and R_4 with R_1 respectively, and then fill out a questionnaire regarding how well each search was performed according the four criteria: Recall of the financial domain component search (R), Precision of the financial domain component search (P), Result Display (RD), and Adaptation Suggestion (AS).

So far, 127 users with 5 years (average) software development experiences had tested the tool. The results are given in Figure 5. It's clear that there is much improvement in R, P, RD and AS in the domain-related MVICS search tool. The linkage between MVICS and financial domain ontology provides a mechanism to represent both generic (CBSE) and domain specific (financial domain) component knowledge in a hierarchical structure and builds relationships to add more semantic meaning, which helps find the most related components. Furthermore, the precision of result components obtained from the financial ontology classes can be calculated on the basis of MVICS classes through the linkage. This should lead to improved R and P. Compared with the SQL database search and the domain ontology-based component search, the RD and AS are new for component specification and retrieval. The MVICS approaches improve substantially in these two aspects.

VIII. CONCLUSION AND FUTURE WORK

The objectives of the research are attained by integrating domain knowledge into the MVICS ontology-based approach as an interlinked domain ontology model. This extension solves the component mismatch problem via both holistic, semantic-based and adaptation-aware component specification and retrieval and extending the search precision and scope by the linkage with domain ontology.

The extended MVICS approach supports dynamic and user group oriented retrieval by adjusting the fiducial facet weights. The presented domain linkage method improves the function and application scope of the component retrieval by linking MVICS to different domain specification ontology. With the linkage, the MVICS model is refined and the repository is extended. A financial domain related software system ontology is built as a case study to explore the possibility of integrating the MVICS model with domain ontology.

Our case studies and user feedbacks have shown that the approach and the tool are promising in tackling the drawbacks in component specification and selection. In the future, we plan to improve the MVICS approach by developing a mechanism for the evolution of MVICS model and its linkage with domain models.

REFERENCES

[1] Baader, F. (2007) "The Description Logic Handbook: Theory, Implementation, and Applications". *Cambridge University Press*.

[2] Braga, R., Mattoso, M., and Werner, C. (2001). "The use of mediation and ontology technologies for software component information retrieval", *SSR 2001*, Page(s): 19-28.

[3] Braga, R., Werner, C., and Mattoso, M. (2006) "Odyssey-Search: A multi-agent system for component information search and retrieval". *Journal of Systems and Software 79*(2): Page(s): 204-215.

[4] Crnkovic, I., Hnich, B., Jonsson, T. and Kiziltan, Z. (2002). "Specification, implementation, and deployment of components", *ACM 45(10)* Page(s): 35-40.

[5] Li, C., Liu, X., Kennedy, J. (2010). "A Holistic Semantic Based Adaptation-Aware Approach to Component Specification and Retrieval". In *Springer LNCS, Tunisia: Springer Berlin / Heidelberg*.

[6] Liu, Q., Jin, X. and Long, Y. (2007). "Research on Ontology-based Representation and Retrieval of Components". *8th ACIS International Conference*, Page(s): 494 - 499.

[7] Ostertag, E., Hendler, J., Prieto-Diaz, R. and Braum, C. (1992). "Computing Similarity in a Reuse Library System: An AI-based Approach". *ACM Transactions on Software Engineering and Methodology*, Vol. 1(3), Page(s): 205–228.

[8] Prieto-Diaz, R., Freeman, P. (1987). "Classifying Software for Reuse". *IEEE Software*, Vol. 4(1), Page(s): 6–16.

[9] Sugumaran,V., Storey,V. (2003). "A Semantic-Based Approach to Component Retrieval", *The Database for Advances in Information Systems*, Vol. 34(3).

[10] Yao, H., Letha, E. (2004). "Towards A Semantic-based Approach for Software Reusable Component Classification and Retrieval". *ACMSE'04*.

[11] Zaremski, AM., Wing, M. (1993). "Signature Matching: A Key to Reuse". *Software Engineering Notes*, Vol. 18, No. 5, Page(s): 182–190.

[12] Zaremski, A.M., Wing, M. (1995). "Specification Matching of Software Components". *Software Engineering Notes*, Vol. 20, No. 4, Page(s): 6–17.