Distributed Healthcare Framework using Patient-centric Role-based Security Modelling and Workflow

Matthew Symes

Submitted in partial fulfilment of
the requirements of Napier University
for the Degree of
Computer Networks and Distributed Systems

School of Computing

August 2008

Authorship Declaration

I, Matthew Robert Symes, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained **informed consent** from all people I have involved in the work in this dissertation following the School's ethical guidelines

Signed:

Date:

Matriculation no:

PLEASE NOTE that in signing this page you are aware of the consequences of doing this fraudulently as explained at http://www.napier.ac.uk/ed/plagiarism/homepage.htm

Data Protection Declaration

Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below *one* of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Contents

Figures

Tables

Acknowledgements

I would like thank Professor William J. Buchanan (Bill) for providing guidance and being my supervisor, Professor John Kerridge for agreeing to be my second marker, and Napier University for providing learning resources. I would also like to thank Napier University and Microsoft for facilitating the Academic Alliance program, without which, this project would not have been possible. In addition, I would like to thank my employer for allowing me the time to complete this project. To my partner I thank for her enduring patience and encouragement. I would also like to thank Dr. Christoph Thuemmler, consultant physician and a Visiting Reader in the School of Computing, for his participation in the evaluation of this project. Thanks also to Liz Binnie who kindly provided transcriptions of recorded interviews used in this document and Jonathan Miles for proof reading.

Abstract

Healthcare professionals are spending less time with patients and more time on administrative duties (Royal College of Nursing, 2008). This is due to a high bureaucratic demand (Brindley, 2007) on the caring process, patient population and longevity (Fougère and Mérette, 1999). A patient-centric system uses gathered information and includes the patient in its functional design (IBM, 2006). Patient-centric requirements have existed in UK healthcare IT since 2000 (Fairway, 2000). Some existing systems cannot be patient-centric. This is because the strategies that shape the requirements for IT systems have changed over time (Mackenzie, 2004); therefore, information technology solutions, built in different times, meet different healthcare requirements. The data created in these differing systems can become disparate and less useful (Singureanu, 2005). Patient information is sensitive; medical healthcare professional roles, such as doctors can only access a patient's health record at appropriate times. Other healthcare professionals must ask for a patient's permission to access their health record whilst other roles in the National Health Service (NHS) are only entitled to non-medical information (Scottish Consumer Council, 2007). This implies that viewing patient data attributes are only permissible by role.

The aim of this project is to provide a patient-centric prototype distributed system that can demonstrate approaches to reducing complexity through data and interface integration; increasing visibility through relevant role based information targeting; and reducing administrative overhead through electronic workflow.

This report examines the history of IT strategies in the NHS, identifying some of the key aims from 1992 to 2008. It then discusses some of the standards defined to allow differing systems to communicate and highlights some of the existing IT systems in healthcare today.

The system design allows patients to interact with the in same way as healthcare professional, it provides access to personal space that displays tasks for the patient or healthcare professional to complete. Data integration is used to build a patient record from local and disparate data sources. Information targeting allows the patient or healthcare professional to visit an area that only displays information relevant to the person there. Finite State Machine methodologies are used to design an electronic workflow, which maps a business process of making a referral.

Using the Microsoft Office SharePoint Server information management framework, data integration is achieved through XML definition and the gathering of meta-data (Hoffman and Foster, 2007). Information targeting is achieved through personalised filtering and security permission modelling (Holiday *et al.*, 2007); workflow is accomplished through the application of design; manipulation of the framework and dedicated workflow (Mann, 2007) code libraries built on top of popular ASP.NET web server technology (Walther, 2006).

This project finds that whilst it is possible to implement these approaches in theoretical context, more research is required into the application of such approaches in real world scenarios. In addition this report finds that software boundaries within the framework suggest the capacity for vast record user, security and management (Curry *et al.*, 2008), however, research into the factors that affect these boundaries, such as concurrency and healthcare professional/patient activity on such systems, is required in order extrapolate accurate scalability requirements.

1.Introduction

1.1. Introduction

This chapter outlines the justification for this project by providing context through the analysis of issues faced by professionals in both healthcare and IT. The main deliverables of this dissertation are defined in this chapter including a list of the key objectives to be completed. The dissertation structure in summarised at the end of this chapter; this is inclusive of a description of the limitations imposed.

1.2. Context

Healthcare Professionals (HCP) are faced with increased bureaucratic responsibilities, which takes up more of their time. HCPs comprise of doctors, nurses and other related health care roles. 85% of approximately 1700 nurses indicated that they could spend more time with patients, if they had help with paper work, (Royal College of Nursing, 2008). This opinion is also reflected by General Practitioners (GP)s who have left the NHS due to being made to sacrifice patient care for bureaucratic procedures (Brindley, 2007). Other factors that contribute to a high volume of work in general for HCPs in most of Europe are population growth and higher longevity amongst the total population (Fougère and Mérette, 1999). In short, more people are living for longer and older patients account for an increasing proportion of the total medical care available. Ironically, improvements in healthcare are partly responsible for this phenomenon.

NHS IT systems are not designed to meet the same goals. This is because patient-centric approaches have been introduced to NHS strategies over a period of time (Mackenzie, 2004). The patient-centric approach is defined by the gathering of useful information using existing technologies combined with the explicit inclusion of patients in systems. This results in benefits for both the organisation; and the patient in terms of efficiency (IBM, 2006). Patient-centric approaches to build on IT systems have been outlined in UK by government strategies since 2000. Many of these strategies aim to empower the patient through inclusion, self-control and delegation of patient responsibilities. Implementations of these strategies are starting to become a reality through systems such as Oasis (Balsam Healthcare Corporation, 2006). Other systems, built on more mature principles (NHS Executive, 1998) and communications standards (Nation Electrical Manufacturers Association, 2007), cannot be patient-centric, alone. These earlier, well-established systems and standards perform well in their designed domains, but, require consolidation with other systems, through additional communications standards in order to be useful in a patient-centric way (Indrajit and Verma, 2007). Visibility is proportional to the availability of information across an organisation. To increase visibility is to make more information from more sources available for consumption (Telemedicine, 2008) therefore; making the information gathered in a patient-centric system more available to users, increases visibility.

Some HCP roles may require attributes of patient information that may not, either by policy or by law, be appropriate for others to access; for example: a hospital porter does not need to know a patients' address, whereas an appointments clerk would require it to fulfil their

role. Patient information security is paramount within the NHS and the professional bodies that comprise of HCP memberships. The NHS consists of many different professional roles, each of which, require a different level of access to patient information. In IT a role is defined by levels of access or groups of permissions that allow access to various objects or views of objects. One example of permissions may be "Read". A read permission may be given to a role and subsequently assigned to an object (Allen and Lowe-Norris, 2003). In terms of an IT problem, the NHS represents a vast pool of information objects in different locations. When examined in a HCP role-based context, parts of the pool are relevant to some HCP roles irrespective of physical location. Some systems like tQuest ordering system from Indigo 4 Systems (Indigo 4, 2004), offer role-based security in specific terms, however, a consolidated security model imposed over differing system designs for location and architecture can be more difficult to achieve (Davies, 2008).

Strategies implemented by the NHS, so far, have failed to meet their objectives within the specified delivery schedules. Slippage was the main cause for some of the proposed systems, e.g. NHSnet, Patient Administration Systems (PAS) and email systems (Greening, 2004). Connecting secondary care systems to primary care trust; i.e. GP systems to Hospital systems, was originally a target for the year 2000 as defined in the 1992 & 1998 Information Management & Technology (IM&T) strategies (NHS Management Executive, 1992) (NHS Executive, 1998). However, by 2005 its inclusion in NHS Connection For Health's (CFH) list of deliverables indicates that this target was not met. Another cause is that both NHSnet and the National Programme for IT NPfIT suffered from a loss of confidence from users and professional bodies (MillarHutchinson, 2000), (Hendy *et al.*, 2005). The success of any system in the NHS depends significantly on the attitude toward it and NHS Care Record System (NHS CRS) systems have been portrayed in the news as putting patients at risk (BBC, 2006). In addition, two of the major IT suppliers for the NPfIT have has their contracts terminated: Fujitsu implementing NHS CRS (Young, 2008); and EDS running NHSMail (Greening, 2004). A further supplier has pulled out of the project: Accenture Consultants (Bowers, 2006).

1.3. Aim and Objectives

This dissertation aims to provide a working, tangible distributed system prototype incorporating methodologies that address the core issues discussed in Section 1.2. This will be achieved by completing a number of key objectives:

1. Reduce complexity by:
    - Integrating disparate data in a useful and reusable way.
    - Integrating interfaces for information consumption.
    - Reducing training diversities.
2. Increase visibility and relevance through role by:
    - Focussing information.
    - Defining information access rights based on HCP and patient roles.
    - Extending reach through integration.
3. Reduce administrative overheads by:
    - Reducing paperwork.

- Reducing effort required to complete a business process through automation.
- Reducing risk in communication and error.
- Implicit inclusion of patients in relevant processes.

Each of these objectives will be applied in a patient-centric context; adhering to the latest NHS strategy.

To facilitate the accomplishment of these key objectives, there are a number of practical objectives including: the procurement, installation and configuration of a supporting infrastructure; the installation and configuration of a core Information Management (IM) framework.

1.4. Dissertation Structure

This dissertation will begin with a literature review that discusses the requirements, strategies and standards which shaped the evolution of modern IT systems in healthcare. The literature review will also look at a small selection of implementations used in healthcare today. An high level analysis of the system used in the prototype deliverable will follow; finishing with a detailed examination of the framework components used for implementation. The middle section of this dissertation in concerned with practical design and implementation of the prototype system; and how each design element addresses the objectives discussed in Section 1.3. The methodology of the prototype implementation will also be documented along with descriptions of any key challenges, decisions made, reasoning and tradeoffs. The final sections include an evaluation of the produced system in terms of its ability to meet the key objectives of integrating data, relevance through role and reducing administrative overhead; this includes comments and analysis from IT experts and a HCP/Healthcare researcher. The core text of this paper ends with conclusions including suggestions for further work.

Due to data protection, no real patient data was used. This project does not set out to provide a definitive solution using real world data, systems or processes, and provides some suggested methods on which practical applications could be built. Any NHS business processes are assumed based on experience.

2. Literature Review

2.1. Introduction

This chapter discusses the history of IT systems requirements in the health service and examine some of the systems implemented to meet these requirements. It begins by examining some of the key strategies, standards and protocols influencing the continuing development of IT systems within the NHS; describing historic strategic changes and the extent to which the principles outlined in these strategies have been implemented in IT systems. This discussion leads into the reasoning for the decisions made in the technology choices for this project.

2.2. NHS IT Strategies

This section looks at the history of strategies and government white papers concerned with the development of healthcare systems in IT.

2.2.1. Information Management &Technology Strategies 1992, 1998

The development of IT systems for healthcare services can be compared to the approaches used in modern western medicine in that both approaches are symptomatic. In terms of IT, system solutions were created to serve specific symptoms or requirements for the NHS. While these systems may perform well in their own right, they are often old, proprietary, designed for a specific purpose and are not easily interoperable. This is underpinned by the findings of the 1992 Information Management & Technology (IM&T) Strategy of the NHS Executive where the *"…lack of integration and commonality…"* between IT systems was found to inhibit progress (NHS Management Executive, 1992). The 1992 IM&T strategy outlined a number of key principles that together formed some indication of the direction the NHS executive required IT systems to be heading. Some of these aims included:

- An NHS wide network facilitating the sharing of information (NHSnet).
- A number representing each patient.
- Systems integration in order to reduce repeating data.
- Security and confidentiality of information.
- Extracting management information from daily tasks.

The 1992 strategy was implemented at local level and nationally supported; it was said that too much emphasis was placed on deriving management information from day-to-day tasks (Mackenzie, 2004), and was replaced by the 1998 strategy. Included in the new 1998 strategy government white paper was "Information for Health 1998" (NHS Executive, 1998). Information for health was a seven-year implementation that outlined the same key aims as the 1992 strategy, but was more specific in terms of those aims which included the National electronic Library for Health (NeLH) and specifics on providing information to patients and the public. The National Health Service Information Authority (NHSIA) was the organisation charged with overseeing the implementation of the strategies (Urquhart *et al.*, 2001).

2.2.2. The NHS National Plan

In July 2000, a new white paper was published: The NHS Plan included several enhancements to the already running 1998 strategies as well as criticism of the IM&T, in terms of investment and NHS trust uptake (Department of Health, 2000). The plan included a number of specific IT expectations including:

- Test results for patients via email.
- NHSnet connection for all GPs by 2002.
- Patient Access to Electronic Patient Records (EPR)s by 2004.
- Electronic outpatient appointment booking by 2005.
- Electronic referral recording.
- Electronic medicine prescribing by 2004.

The NHS Plan was the first to suggest a patient-centric approach, redefining the NHS around patient requirements (Fairway, 2000). It identifies the issues discussed in Chapter 1 through its IT requirement strategies. Patient access to EPR identifies the need to integrate disparate EPR data into an interface that displays relevant information based on the patient accessing the system. In terms of reducing paperwork, the implementation of such systems as electronic booking and referral recording could address this issue by consequence.

2.2.3. eGovernment Interoperability Framework (2000)

This strategy acknowledged the dominant Internet standards by requiring all new NHS system developments along with existing systems to be compliant, supporting such mark up notation as eXtensible Markup Language (XML).

2.2.4. The National Programme for IT (NPfIT) / NHS Connecting for Health (NHS CFH)

NHS CFH was formed on April 1st 2005 for the English NHS signifying the end of the IM&T seven year plan and the closure of the NHS Information Authority (NHSIA). It was charged with a delivery of the NPfIT objectives. NPfIT is built on the 1998 IM&T and the NHS Plan. Included in the programme were the following objectives: connecting GPs to hospitals, a central email system, GP-to-GP record transfer, Picture Archiving Communications Systems (PACS), Chose-and-book electronic booking service and a system for electronic prescriptions. Some of these NPfIT deliverables, by their very inclusion in the programme, indicate that many of the goals of IM&T and the NHS Plan were not met within the respective time periods.

Today NPfIT is running into similar difficulties as the 1998 IM&T strategies did. These are a loss of confidence in system security and change found by two studies: Hendy et al (Hendy *et al.*, 2005), in their interviews with NPfIT project managers and regional NHS Trust directors and the Miller Hutchenson report on NHSnet in 2000. This represents a seven-year period in which these issues were still not addressed. More recently, implementations of the NHS Care Records Service (NHS CRS) have been criticised for putting patients at risk (Foot, 2008), (Doward, 2008).

Between the publication of the NHS National Plan and NPfIT, network infrastructure changes improved communication through broadband connections and NHSnet became N3. Furthermore, one significant addition to the NPfIT was the NHS CRS, this is discussed in Section 2.5.3.

2.3. Standards and Protocols

This section looks at selected standards and protocol definitions used in healthcare care systems today. It begins with a look at image messaging standards followed by analysis of broader messaging standards and workflow terminology.

2.3.1. Digital Imaging and Communication in Medicine (DICOM)

In the 1970's, following the invention of Computer Tomography (CT) and other digital imaging techniques, the American College of Radiology (ACR) and the National Electrical Manufacturers Association (NEMA) recognised a requirement to standardise the way in which information was exchanged between these devices. ACR-NEMA version 1 published in 1985 comprised of a set of standards which aimed to *"promote communication of digital image information, regardless of device manufacturer" (Nation Electrical Manufacturers Association (NEMA), 2007)* as well as facilitating archiving and improving information distribution through diagnostic database implementation. Published in 1992, DICOM was the third revision of the original ACR-NEMA standards. DICOM standards describe all the services, data structures, syntax necessary to establish interoperability with other DICOM compliant devices. The DICOM networking standards require that communication is established over Transport Control Protocol/Internet Protocol (TCP/IP).

With DICOM, Service Classes (SC)s are used to expose functionality between devices. The services pertaining to distributed networking are: store; query; and retrieve. It defines a client-server model through Service Class User (SCU) and Service Class Provider (SCP) annotations. Images and Information such as patient data are stored in a single data structure known as an Information Object (IO) so that a test or study image and patient ID cannot be separated or mistaken. Larger DICOM implementations use shared archives from which IOs are retrieved (Parisot and Solomon, 2005). Figure 2.1 shows a DICOM network topology indicating SCs used to store and retrieve an IO across a network.

DICOM standards do not describe a language construct; therefore DICOM standards can be implemented using languages such as C#, C, C++, Java and Python (LeadTechnologies, 2007). It is not an architecture or a framework, and alone, cannot guarantee interoperability (Nation Electrical Manufacturers Association (NEMA), 2007). In addition, the DICOM standards are concerned with medical informatics only; other aspects of patient care are not taken into account and as such cannot be patient-centric. However, DICOM defined data structures can be used in a patient-centric way through integration with other information systems. DICOM is currently implemented in almost all medical imaging systems therefore any design of an overlying information management system must take these established standards into account.

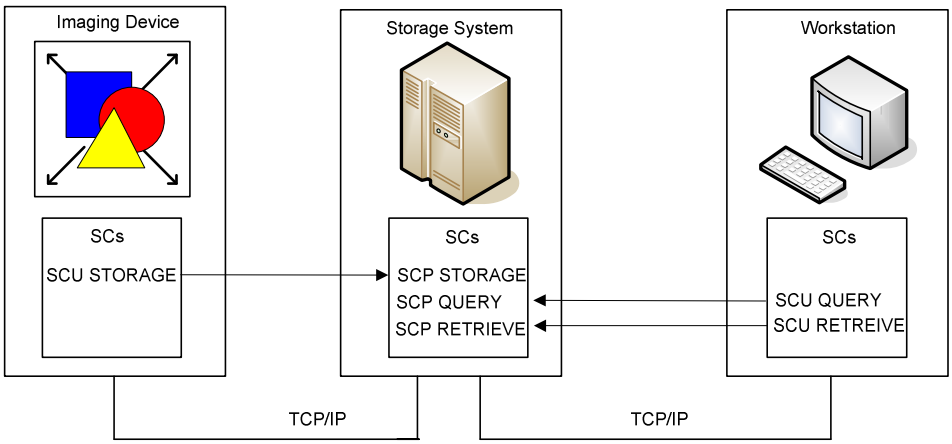


Figure 2.1: DICOM Distributed Model and Service Classes

2.3.2. Health Level 7 (HL7)

HL7 is an organisation that participates in the development of healthcare standards in IT. Such standards set out the language guidelines and methodologies required to allow compliant medical information systems to exchange information. HL7 standards are divided into component parts that tackle the many facets interoperability. HL7 v3, released in 2005, is the latest set of HL7 standards, and some of the HL7 components are discussed in this section.

2.3.3. Reference Information Model (RIM)

The RIM defines the rules by which distributed IT heath care systems communicate. Many countries use different medical terminology to describe the same thing:

> *"Computers cannot deal with synonyms (saying the same thing using different words) or homonyms (where the same term or phrase means different things depending on context)" (Benson, 2007)*

HL7 v3 defines standards by which medical terminology is communicated in an unambiguous way through unified coded representations of medical terms which are referenced from the RIM. The RIM also specifies permitted elements, constructs and relationships for messaging protocol, data types and attributes. Events in healthcare are defined in the RIM as "Acts" or "happenings". In real terms an Act may include medical activities such as procedures or medications as well as others. Context is given by what HL7 defines as "Participation". For example a "performer" participant may perform a procedure Act and in the same Act, a "subject" participant could also be included. The participators are defined in the RIM as "Roles", and a Role can include anything from doctors and patients to facilities and samples. Extending the previous example; a patient Role can be the subject Participant of a procedure Act and so on. Roles comprise of RIM "Entities", these include organizations, people, devices, etc (Amy *et al.*, 2003).

2.3.4. HL7 v3 Messaging

Messaging in HL7 v3 is accomplished through XML files. Permitted XML tags and attributes, along with an attribute's specified data type, are derived from the RIM; XML schema documents are used to define which tags and attributes are allowed in a message file, their order and occurrence.

2.4. Workflow

According to businessdictionary.com, a workflow is a *"Progression of steps (tasks, events, interactions) that comprise a work process"*. Workflows are inherent through the NHS as treatment can be examined as a business process in almost every field of medicine from accident and emergency (Malamateniou *et al.*, 2003) to radiology (Karlsson and Eklund, 2001) Electronic workflow can reduce the requirement of paper based processes. It can also reduce human error by removing possibilities of error making from the business process. This ensures information associated with a business process is in the right place and with the correct permissions for the state of the process. This can increase patient efficiency, reduce administrative overhead and free up time for HCPs to care for patients (Ball et al., 2003).

Workflows in medical healthcare can be service orientated and consent-based. Service orientated workflows are exposed via web services to allow portability and range. Consent-based workflows can be inclusive of patient interaction; not only allowing the patient to opt-out of a workflow but also determining in which situations this is not possible (Russello, 2008).

Both DICOM and HL7 v3 were amongst the many initiatives created to improve workflow. Both offer classes, objects or attributes to allow workflow to be accomplished. Both avoid stipulating exactly how to manage such elements in terms of the real world implementations (Indrajit and Verma, 2007). DICOM has a rudimentary workflow interface that can receive patient data from a Hospital Information System (HIS) in the context of scheduled work objects. The attributes defined in HL7 v3 RIM can allow for much more granular workflow connotations including temporal constraints (Singureanu, 2005).

2.5. Past and Present Implementations of NHS IT standards and strategies

This section discusses IT implementations in health service organisations, their relevance in terms of strategy and how they compare to the objectives discussed in Section 1.3. The section begins with a look at the systems implemented as result of the IM&T 1992/1998 strategies, followed by some analysis of more recent implementations.

2.5.1. NHSnet

NHSnet (now N3) was the networking framework system that linked all primary and secondary care bodies of the NHS. It provided services that, in turn, provided platforms for accomplishing most of the aims of the 1992 and 1998 strategies. This includes internal and external web pages to allow HCP and patient access respectively, policies and mechanisms by which patient identifiable information was secured and support for electronic patient records. These implementations were controlled and regulated by the then new National Health Service In-

formation Authority (NHSIA). Described as the “mini internet” within the Internet (Dobson, 2000); connection to NHSnet was originally accomplished via desktop application through a number of proprietary dial-up connection protocols. The communications infrastructure was installed and run by a combination of BT and Cable & Wireless communications companies. Each connection to NHSnet was charged per minute with additional charges for hiring routing equipment. An overview of the NHSnet topology can be seen in Figure 2.2

Plans to connect all GPs to NHSnet by the end of 1999 were shelved. One reason is that the British Medical Association (BMA) warned its members not to sign up for the service, due to concerns over security. Subsequently, some publications reported that only a third of GPs signed up to the service, three years after the service began (Dobson, 2000). Other issues with NHSnet in 1999 included: poor internet access speed; confusion over roles and responsibilities; and poor support from both NHSnet and core suppliers such as BT (MillarHutchinson, 2000).

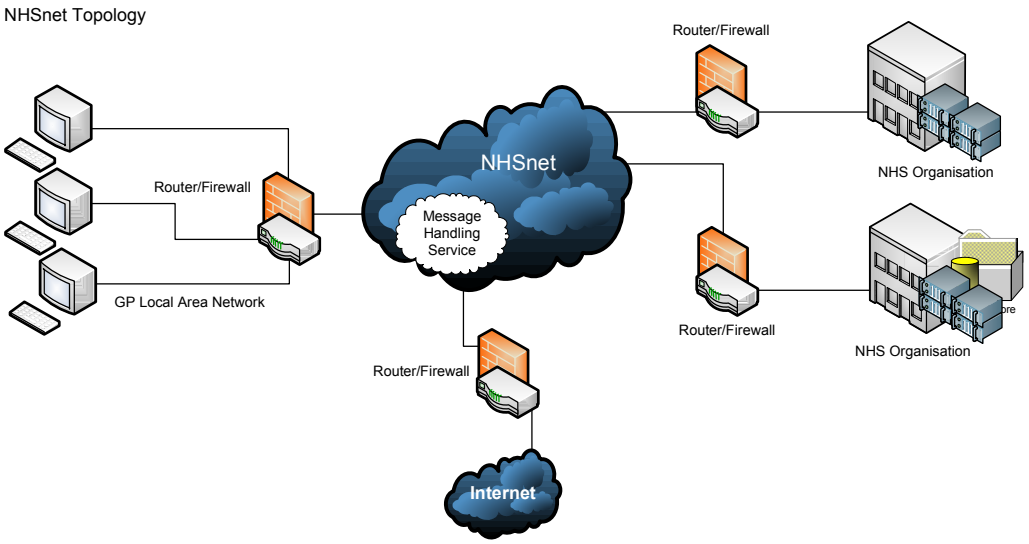


Figure 2.2. Example of NHSnet Topology

2.5.2. National electronic Library of Heath (NeLH)

The NeLH is one realisation of the 1992 IM&T strategy. It consists of a set of services concerned with the retrieval of medical information which includes images, journal findings and case-based information. Users access the NeLH from web-based browsers and are allocated services based on user rights. Originally modelled on a real library, NeLH provided a main virtual entry point with sub libraries devoted to particular fields of medicine. Users were expected to visit these sub-libraries virtually from an initial landing page. In April 2008 NeLH introduced a new search engine NLH 2.0 that would move away from this approach by providing a one-stop interface to all information (Evans, 2008).

2.5.3. NHS Care Record Service (NHS CRS)

NHS CRS is a clustered data record service representing five regions in England. It was designed to make patient data, including treatment history and images, available centrally to HCPs and patients. Data is obtained from two levels, national and clustered. National level

provides some booking functionality and a summary patient record obtained from a central database known as the Spine. Cluster level gives access to a wider range of services (Hendy et al, 2005) and a more comprehensive patient record held locally at the place of treatment. Cerner Millennium Care Records Service is example of a NHS CRS. It is installed in several Primary Care Trusts (PCT) across England. In 2007, Milton-Keynes PCT reported a successful transition to the system (Clark, 2007). By contrast, its installation into the London PCT in 2008 is said to have caused chaos and risk of patient safety (Colins, 2008).

2.5.4. Indigo 4 Review and tQuest

Review and tQuest by Indigo 4 are two commercial implementations of Health Information Systems. Combining these with other Indigo products and third-party collaborations forms an information management framework for requesting testing services such as: radiology; pathology; phlebotomy management; and lab-to-lab requests and reports. Requests for tests are made through Indigo 4's tQuest. The system's interface is browser-based and is capable of supporting primary and secondary care scenarios by connecting GPs, wards, specialists and laboratories. The system has role-based security to prevent overspend on expensive tests, uses a Patient Administration System (PAS) and or a Master Patient Index (MPI) to obtain patient information. PDAs can be used as a result of collaboration with third parties. Review is the reporting part of the system and the results interface. Workflow only operates at the primary care level within the system. However, the status of a request can be monitored through Review (E Health Media, 2005). tQuest and Review can provide role-based user access rights and information to patients. However, it cannot integrate other data from laboratories or hospital departments using different systems (Indigo 4, 2007). tQuest and Review also have workflow mechanisms, but these do not cross system boundaries and are not completely patient-inclusive.

2.5.5. Indigo 4 Keystone

Keystone is another product by Indigo 4. Keystone is a message broker, it is compatible with the main standards of healthcare messaging including XML and HL7 (Indigo 4, 2004). Keystone offers message encryption and standards translation between source and destination. Keystone seems to lie inside the same information domain as Review and tQuest yet it is unclear from the available literature if Review and tQuest require Keystone to function. Message brokers like Keystone facilitate communication between legacy systems. This saves money on retro fitting existing systems to meet the requirements of the eGovernment Interoperability framework discussed in section (SEC). Keystone reduces paper work, but only for its designed application domain. Keystone integrates data through message translation but is not concerned with how the data is used by the user application.

2.5.6. BHC Oasis Patient Administration System (PAS)

Oasis is a PAS developed by Balsam Healthcare Corporation (BHC) and ELAN in 1994. It is installed in a number of large hospitals in the UK. Oasis is a client-server based technology built on a web server platform allowing access from any supported browser technology. It uses Oracle database products to drive the web-based platform. It does not use HL7 compliant messaging for internal system communication, although compliance can be achieved

through an additional HL7 integration engine. Unlike Indigo 4 Systems, Oasis was developed to span across information domains, to manage any kind of healthcare process through implementation of preconfigured modules geared to handle a particular facet or functional requirement of healthcare. Examples of modules include “Operating Theatres and Anaesthesia” and “Cardiology”. The implementation of modules can be suited to a given installation environment, i.e. if a hospital does not contain an operating theatre, there is no need to implement that module (Balsam Healthcare Corporation, 2006). Oasis is available in two languages. However, it is not known if these language configurations are interoperable across installations.

2.5.7. Oasis Medical Solutions

Oasis Medical Solutions is a UK company originally known as Elan Equipment. Elan was involved in the original Oasis project in 1994. After the installation into Queen Margaret Hospital in Scotland, Elan and BHC developed Oasis separately. Oasis and BHC still continue to have a working relationship today (Pusey, 2008). Both platforms are closed, and modules are preconfigured in the software house and then as part of a solution. The NHS has to rely on these companies to both stay in business and work with them in applying business processes. This can produce an undesirable lock-in for both the NHS and its suppliers (Wray, 2006). In these terms, using an abstract platform increases the size of the skills base from which to create solutions, by allowing the NHS to choose from a much wider range of solution providers.

2.6. Conclusions

It is proposed to develop a system which addresses the core issues identified in Section 1.3. These involve:

- the reduction of complexity through data and display integration
- the increased visibility through role-based, relevant information
- reduction in administrative overheads through workflow.

The issue of disparate data sources can be mitigated by integration. The majority of these data sources can be accessed using DICOM and HL7 messaging standards. These standards define integration, but do not define mechanisms of implementation. DICOM deals with a narrow information domain as does Indigo 4’s tQuest. Oasis offers consolidation of information management domains, but does so on its own terms and does not allow full patient interaction, and therefore does not seem patient-centric. In terms of a patient-centric system none of the discussed systems appear to implement workflow outside of inter HCP practice. Patients are included, but, in many instances as a subject of an end result. Patients are often not thought of as an integral part of the system and are not even considered as users in the case of DICOM, HL7 and Indigo 4.

Visibility and relevance of data can be greatly increased by using two approaches. Firstly, to integrate and federate databases and content sources. Secondly, to focus the relevant resultant information by HCP role and present it using a single interface that requires a single login process.

The use of paperwork can be reduced by using workflows. Logging each stage of a workflow can reduce the administrative overheads because the manual process of auditing is removed from the HCP's tasks. To implement such a system that could meet the main objectives to form the main deliverable in the time frame provided, it was unreasonable to start from scratch. A more realistic expectation was to use a framework upon which this prototype could be realised. Microsoft Office SharePoint Server 2007 (MOSS) was chosen for this system because the framework provides the required functionality to complete the main objectives of the project within the given time frame.

3. Analysis

3.1. Introduction

This section discusses some aspects of MOSS, however, covering this entire product is impractical in terms of brevity, therefore a full overview of MOSS, including additional theoretical applications in a medical environment can be found in Appendix 2. This section discusses only the components of MOSS that were used for the designs and implementations discussed in Chapters 4 and 5, respectively. MOSS is a database-driven information management product which allows manipulation through customisation and development of a publicly available object model. It can be installed and run on some existing NHS trust infrastructures without the need to change or reconfigure any of the components, such as existing IT systems. In addition, MOSS comes with workflow mechanisms that allow a developer to create custom code based on real business processes (Mann, 2007). It contains audience targeting mechanisms that work directly with a required Directory Service in a role-based context (English, 2004). Overall, MOSS is designed for implementation at an enterprise-level; this suggests that it may cope with very large numbers of records more efficiently that current smaller implementations. Finally it also contains a feature which allows consumption of any data from any SQL compatible database or XML web service via an XML-based component known as the Business Data Catalogue (BDC). Napier University was one of the first universities to pilot the use of the BDC in a production environment when they integrated student details from student records database to increase visibility for students in 2006 (Silversands Consultancy, 2007).

3.2. MOSS Service Components

The main MOSS service components are:

- **Index Server**. The index server is used for gathering information various content sources. It uses the same technology as some of the large internet search engines.
- **Web Server**. Standard ASP.NET 2.0 front-end web server used for interfacing with the system. Patients and HCPs access the system through a web browser (Walther, 2006).
- **Query Server**. This is the mechanism by which search requests are received, processed and delivered.
- **Excel Calculation server.** This component provides extended, centralised calculation services to those using Microsoft Excel.
- **Shared Services.** Shared Services consist of a collection of components that handle all user profiles, information targeting, personal web space, search settings and business data connections and applications.
- **SharePoint Web Services.** Almost all the functionality available in both the Windows SharePoint Services (WSS) and MOSS object models can be leveraged through web services. This allows MOSS installations on differing networked domains to communicate.

3.3. MOSS Server Components

Each component of the MOSS system can be located on a separate server or, for smaller environments, located on the same server. This includes the database. A collection of these components running on more than one server in any topological configuration is known as a farm. A MOSS farm requires supporting server components such as an SMTP email server, a Directory Server and Internet Information Services (IIS) web (English, 2007).

3.4. MOSS Software Components

MOSS is built under Windows SharePoint Services 3.0 (WSS 3.0), which is a collaborative web technology. WSS 3.0 sites are expressed through a front-end server component which is basically an Microsoft Internet Information Services (IIS) .NET 2.0 web site. MOSS's structure is made up entirely of WSS 3.0 sites (Holiday *et al.*, 2007). A site can be configured then saved as a template with or without any inclusive data. Just as a MOSS site collection is a structured collection of WSS 3.0 sites, WSS 3.0 sites are made up entirely of WSS lists, so a WSS 3.0 site is a structured collection of WSS lists. A list is a container in which items of information can be stored, shared, managed and secured. They can also be considered as a table in a role-based security enabled database. A WSS list comprises of site columns and content types. Content types are the core information types that make up the framework. An example of a content type is a "Basic Page", this content type used by the system display .NET 2.0 aspx pages. Content types can be based on another content type when created, the Basic Page example described earlier is based on a content type "Document", which is known as the parent content type (Hoffman and Foster, 2007). Site columns allow global use of list columns across the framework. For example; if a custom list is created with a column "PatientID", PatientID may be required for use in another list. Instead of creating an additional column called PatientID for the additional lists; a site column is created and chosen from a list of existing site columns. This can allow HL7 v3 message types to be expressed as SharePoint content types via MOSS forms technology (Kush *et al.*, 2007), providing an interface to HL7 standards similar to Oasis's HL7 interface engine.

3.5. Delivery Mechanisms

WSS 3.0 sites and pages are rendered as .aspx web pages via the Microsoft ASP.NET web server technology. The display of a site is controlled by asp web user interface (UI) controls and ASP.NET 2.0 Web parts which are also an extension of .NET UI controls (Pattison and Larson, 2007). User interfacing is accomplished through any appropriately configured web browser. A Web Part is a floating, collapsible panel of information. A user, when editing a page, can drag and drop web parts onto Web Part zones. Web Parts can be used to express any information from within a MOSS farm. They can also be used to display data, external web pages and email components. Web Parts are fully programmable, developers can create any piece of programming and express it through a web part (Hillier, 2005). In addition, if the programming code is included, web parts can be connected, provider web parts send an item of information to consumer web parts when the page is loaded forming runtime relationships between native and disparate information sources such as DICOM. In conjunction with directory based list security, web parts can provide role based visual experiences. One HCP may

have access to one set information based their role to another yet both would be visiting the same web page reducing complexity by consolidating information location. Figure 3.1 shows a typical request response to a WSS service. When a request is made the SharePoint Internet Service Application Program Interface (ISAPI) filter determines if the request is for a regular ASP.NET web page or for a WSS 3.0 site. If the request is WSS 3.0 related it is passed to a SharePoint Page handler via the SharePoint security model where it is determined if the request is for a direct, precompiled page or a page that requires compilation at runtime. Code requests are handled by the Microsoft .NET 2.0/3.0/3.5 Framework class libraries; each offering a different set of services which go together to make up the full functionality of WSS and therefore MOSS. Examining Appendix G reveals the relationship between the ASP.NET platform and the rest of MOSS Architecture.

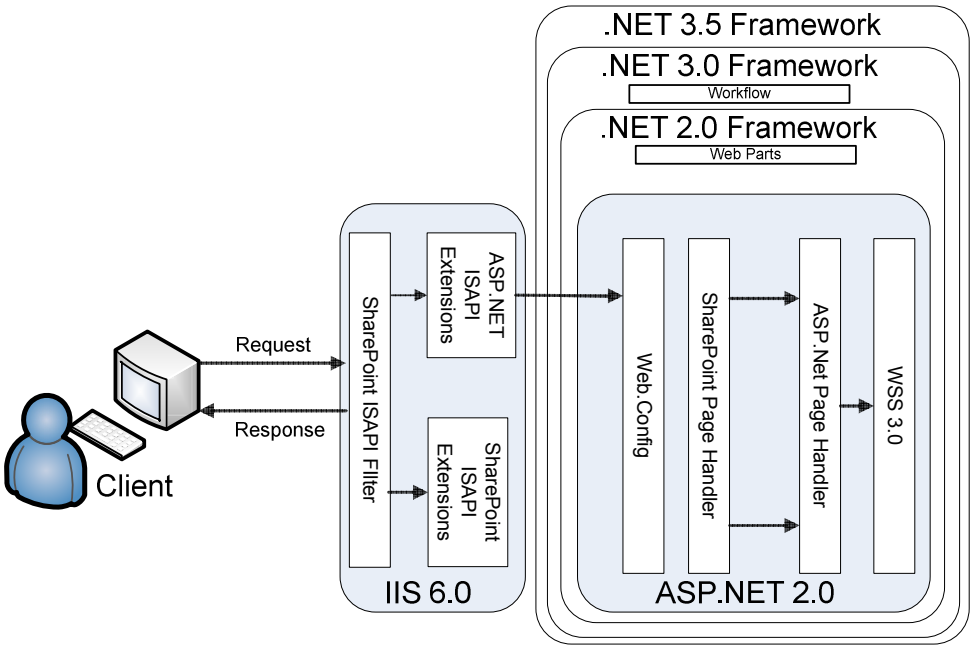


Figure 3.1. Typical request response ASP.NET and WSS

3.6. Business Data Catalogue (BDC)

The business data catalogue is part of MOSS shared services and is used to consume data from disparate sources. Simply using an Open Database Connectivity (ODBC) connection to any data and consuming data directly seems more efficient and is certainly possible, but, the data is then static in the environment, it only means something specific to that requirement at that time. What the BDC does is describe the data as a hierarchical structure or schema allowing the framework to treat the data as if it were part of the MOSS data structure and as so then run methods over the returned BDC data as with a native list. This means the data is reusable in the entire information infrastructure and therefore useful. The BDC can use several types of access techniques found in most databases. Role-based authentication can be used or a single privileged user can be used in a built in single sign on module (Appendix 2) comprising of an encrypted database storing user credentials. Figure 3.2 shows the BDC architecture.

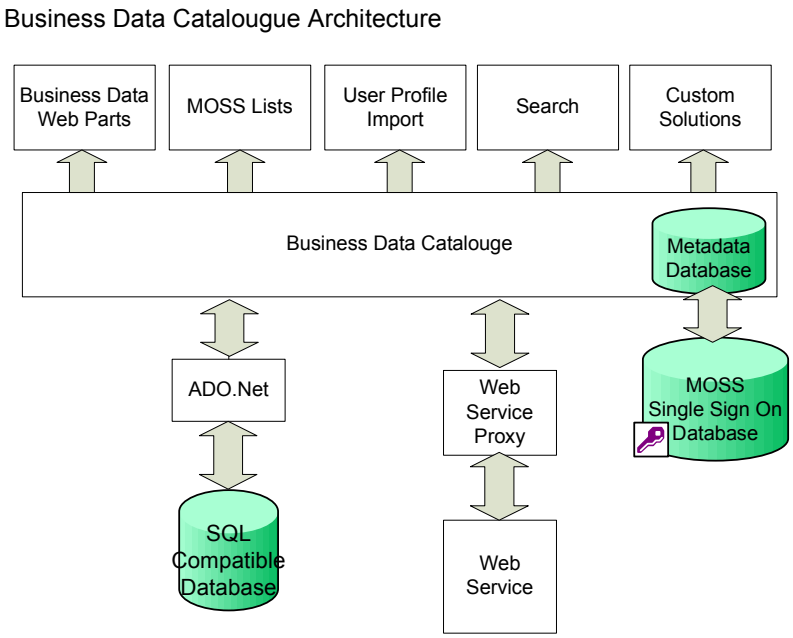


Figure 3.2. Business Data Catalogue Architecture

The BDC uses XML to describe the aspects of the data gathering process. A file known as an Application definition File (ADF) containing the XML is loaded into a part of the BDC known as the Application Registry. When creating an ADF file, the XML used must be valid and correspond to the Metadata container model which is the Object Model used by the BDC to navigate through a described hierarchy shown in Figure 3.3 (English, 2007).

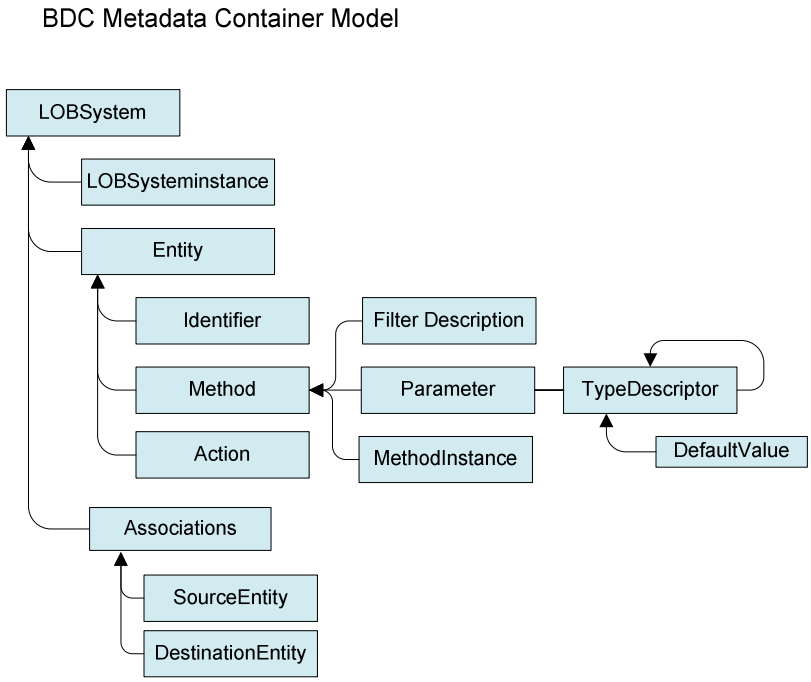


Figure 3.3. Business Data Catalogue Meta Data Container Model

BDC data is available in a one-way fashion, i.e. no write backs, this means that data is only consumed through the MOSS infrastructure and any interaction is on the parent data set. The BDC also interacts with the MOSS's profile database via the indexer, this means that extra attributes from disparate sources such as an EHR at secondary level can be defined and incorporated into user profiles in a primary care context thus integrating data in a useful way (Duerden, 2007). Consuming data from other systems in this way can integrate each source through a single interface reducing complexity for the end user and does not require a change in working process for those operating the disparate data sources.

3.7. Search Service and Indexing

Indexing works by using an index engine to navigate through content sources, gathering Meta data and storing it in a file known as an index volume (English, 2004), this gathering process is known as crawling. Users use a query engine to interrogate the index volume and present results. In addition to crawling MOSS sites, the index engine is capable of crawling other content types such as a Network File Systems and external web sites. Crawler rules can applied to the index engine which governs the extent to which a source is crawled and which aspect, if any, of the source is to be filtered out. Indexing is configured using the search service in a SSP; content sources, crawl logs and rules along with search query configurations such as key words scopes and best bets are all stored in a configuration database (English, 2007). The index engine could be used for crawling NeLH, PACS, DICOM Multipurpose Internet Mail Extension (MIME) Types (Cordonnier, 2002), NHS24 and, HL7 message XML files. In addition, through the MOSS BDC described in Section 3.6 and with permission from vendors, the index engine could crawl, EPR, Electronic Staff Record (ESR) and Keystone Enterprise (Indigo 4, 2004) or any other compatible data source. The most recent and notable use of the MOSS search and indexing was for the Olympics Committee in Beijing (Montalbano, 2008).

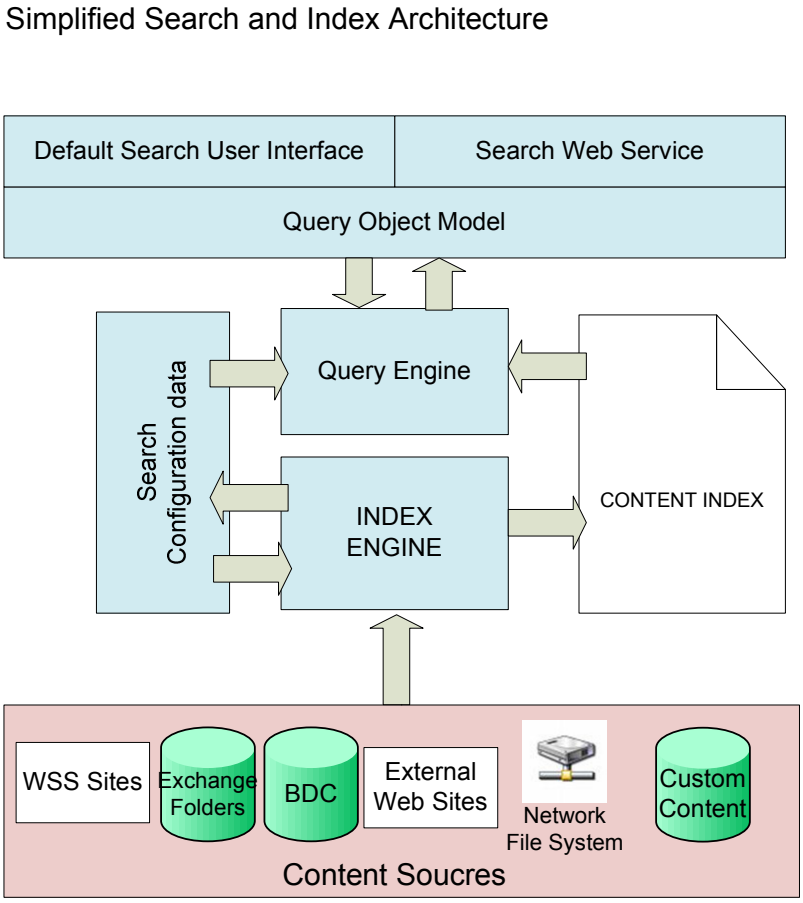


Figure 3.4 Simplified view of MOSS search and indexing architecture

3.8. .NET 3.x Workflow

As stated in Section 2.4, workflow allows business processes to be mapped into electronic format. The .NET 3.0 and, optionally, .NET 3.5 class frameworks are required to run workflow in MOSS although it is noted that .NET 3.0 workflow component can run in any host application (Mann, 2007) such as a DICOM application or HL7 compliant applications, Figure 3.5 illustrates the .NET 3.0 workflow architecture based on a MOSS host application. The .NET framework consists of class libraries which allow the manipulation of the underlying operating system. Workflow consists of several .NET 3.0 class libraries that allow tasks to be created, manipulated and monitored over a time line . There are several standard workflow types available in MOSS without the required programming knowledge allowing users to create simple working practices without the need for code development. There are essentially two types of .NET 3.0 workflows: sequential, this means the work has a beginning and an end, or, state based, this is when the workflow does not end as such, it just exists and changes its state or it does not exist. A state based workflow closely follows the principles of Finite State Machines (FSM) and is created using similar design templates (Freudenstein *et al.*, 2007).

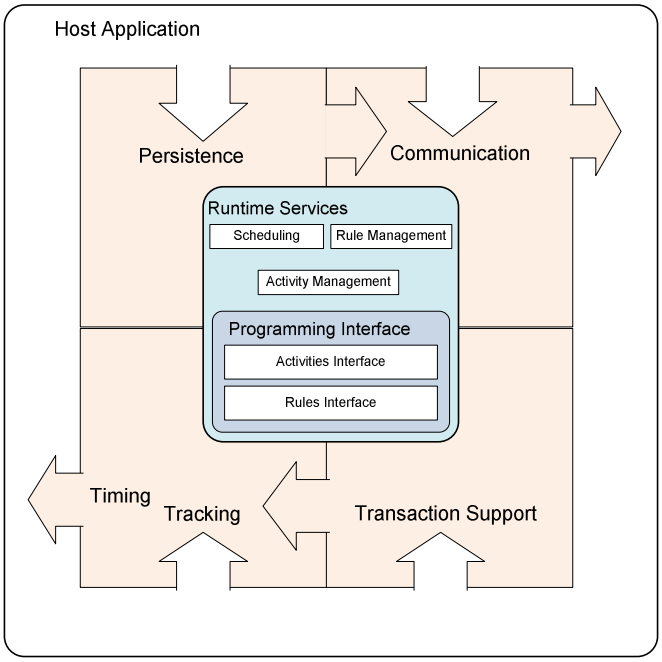


Figure 3.5. High Level Workflow Architecture

3.9. My Site

My Site is the name given to the personal site feature of an SSP. When configured, each user receives a WSS 3.0 site which can contain all of the functionality of MOSS. In this patient-centric system, patients interact directly with the system in this way; a My Site for each patient provides a good platform for displaying EPR or NHS CRS information to the patient through a single interface. The patient is only required to know his or her My Site URL to access information from many different sources, this also reduces complexity. Each My Site comes with web parts that allow the user to see workflow related tasks and documents. My Sites, essentially, consist of two pages, the first can only be viewed by the person to which the site belongs. The second is viewable by everyone who has access to the site. My Sites can be hosted on a separate web application allowing strategic load modelling in farm design (English, 2007).

3.10. Conclusions

MOSS provides extensive functionality in an abstract form that allows a developer to rapidly produce example solutions to a wide range of business needs. The MOSS BDC provides integration capabilities which address the issue of consuming disparate data. Built in display mechanisms allow the developer to concentrate on functionality and the ASP.NET platform provides worldwide access to MOSS through an internet connection. My Site facilitates the patient-centric concept by providing personal web space to the patient complete with task interaction. MOSS workflow provides the platform to realise the objective of reducing administrative overheads through designs that cut out manual procedures in particular, the logging of events.

4. Design

4.1. Introduction

The challenge, in health care, to make information more patient-centric can be solved by design. In this patient centric prototype design and implementation, the patient is required to be a user of the same system as HCPs. This is also the approach of (Hopper *et al.*, 1998) and (IBM, 2006). Patients access the system and use the services through a single interface to discover tasks upon which they are required to act, such as accepting an appointment. HCP's can then view patient record information by visiting the same site. HCPs also have access to specialist departmental areas, such as Surgery and A&E, pertaining to their own professional requirements and the information needs similar to Oasis modules discussed in Section 2.5.6. The system must also demonstrate data integration, and workflow capabilities to complete the main objectives. In addition, it may be beneficial if the system could address some of the wider issues discovered in Chapter 2, thus this Chapter examines the design for the system prototype.

To utilise MOSS, it is necessary to create an enterprise environment consisting of several servers each configured to performed specific roles. In the context of this project, an enterprise environment is defined as a distributed system connected via various network protocols across multiple physical network domains capable of supporting thousands of users, email accounts, databases and web sites.

4.2. High-level design

Figure 4.1 shows the high-level design overview. The system design is a distributed, web-based, client-server system which provides general access to both patient and HCPs through various web technologies such as web browsers, Personal Digital Assistants (PDAs) and mobile phones. WSS 3.0 Web sites are used to represent healthcare departments such as surgeries and laboratories, information such as appointments and laboratory results is stored in WSS lists inside these department sites. List and site contents are secured using a static, directory-based model on privilege through employment role; i.e. administrators may not have access to medical records whereas medical staff may. Underneath this security layer, filtering mechanisms producing targeted information based on access and profile identity. This produces personalised views of information based on the relevance of subject or participation. If a HCP logs on to the page, they see information about patients and procedures they are involved in, patients however, see HCP roles and procedures they are involved in.

An additional web site is allocated to each user; this provides the interface for activities such as viewing a patient record, linking to personalised information pages and interacting with tasks. Patient records are made from different sources consolidated in the system through data integration. A disparate source, representing a data source, such as a NHS CRS summery record, is consumed via a BDC connection achieved through XML-based application definition. The resulting data is integrated with existing local attributes and stored in a profile database. Patients can the view this record through their allocated WSS 3.0 site known as My Site. The system can also use these consolidated attributes for filtering, security and

list functionality. Tasks drive human and non-human activities in terms of the IT system. If a patient has an appointment, the task is that the patient shows up. Tasks have basic attributes such as “due date” for temporal control on a process and “assigned to” for user control; these attributes are stored in WSS task lists. A personalised list of tasks from each departmental site the patient or HCP is a member of, is present on the users’ personal web site. Patients can easily see what is required of them by visiting their personal space. Links in the tasks are connected to documents relevant to the task so that the patient can view associated information instantly instead of waiting for a posted hardcopy. Workflow design in this system automatically assigns users to tasks, changes list information for consumption, logs patient progress and provides email notification. In addition, workflow manipulates the static security model to allow a list item contribution to patients for duration of a task and access to a patient’s personal site for HCPs to view medical history and records for the duration of a case.

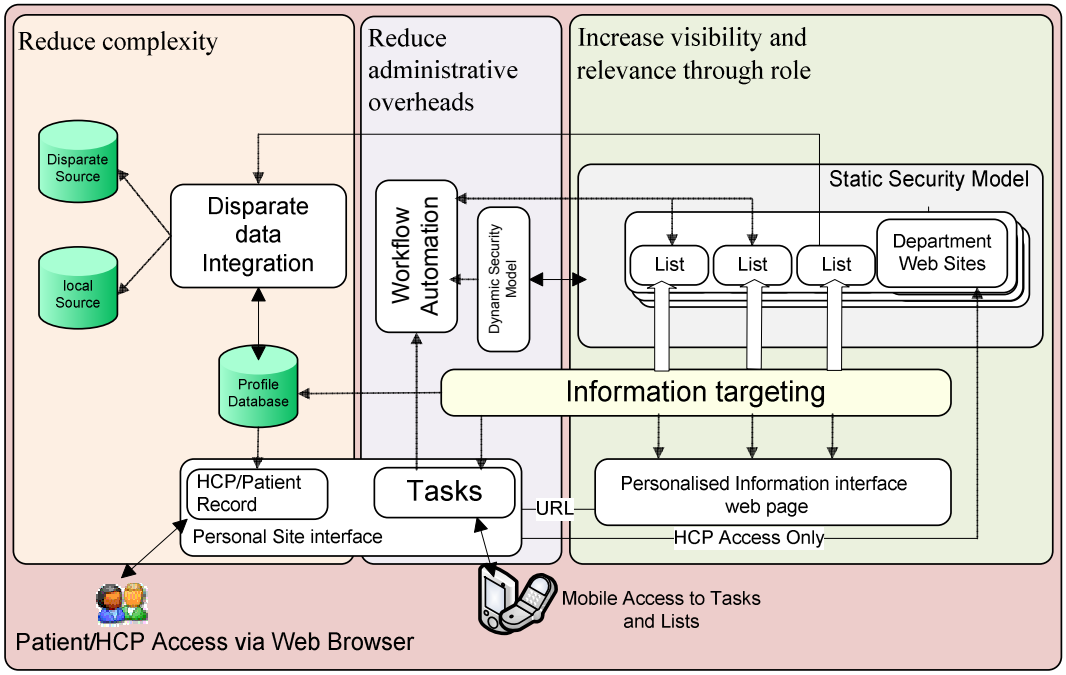


Figure 4.1 High level design overview

4.3. Infrastructure Model

Figure 4.2 shows the main hardware components, intended server roles and topology design. To demonstrate data integration, a PostgreSQL database is located on a network which is accessed via the Internet. The infrastructure design models a distributed system over a Local Area Network (LAN). To allow for comprehensive scalability, the design requires four Windows Server operating systems occupying separate network nodes. Two of these systems are used for a MOSS farm, the remainder for the systems upon which MOSS is dependent. The model design comprises of the hardware and software components required to build the designed implementations in order to meet the objectives discussed in Section 1.3.

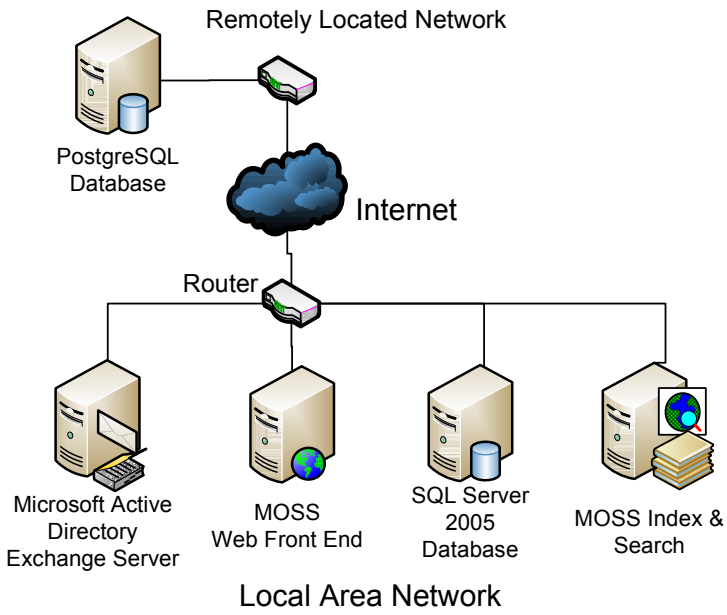


Figure 4.2 Basic network topology and MOSS farm infrastructure design

4.4. Site Structure Model

The site structure model is based on healthcare service departments, where each department is represented by a WSS 3.0 site. Each site is at an equal hierarchical tier in the structure. My Site technology is used for patients and HCPs, where the collection resides on a virtual tier called "/Personal". Figure 4.3 shows the site structure model including the departmental site names used and the lists discussed in Section 4.6.

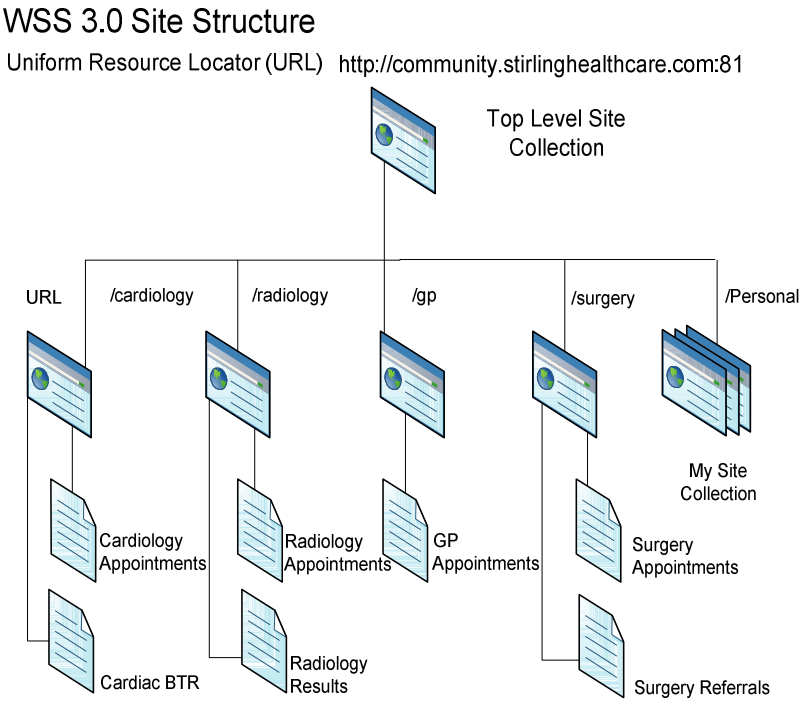


Figure 4.3 Site structure design

4.5. Site Column Definitions

Site column definitions allow information elements to be reused inside the overall design. MOSS has a range of predefined site columns applicable to abstract framework functionality. In order to extend that functionality into something useful for medical application, relevant information elements must be identified and mapped into custom site columns for use in list design. Table 4.1 shows the resultant custom column design.

Table 4.1 Additional site columns

Column Name	Description
Date Tested	Used as time stamp for a testing activity such as a blood test
HCDepartment	List of Departments
Additional Notes	Test element used for adding additional information to a list such as special notes for testing or appointments
Appointment Time	Used for appointments of any description
HCUID	This site column is used for storing a HCP user
PatientUID	Used for storing a Patient userID
PatientID	Test representation of the same patient userID, this column is not subject to the same dynamics placed on patientUID column.

4.6. WSS 3.0 List Templates

A WSS 3.0 list template is used to achieve the objectives of relevance-through-role and a reduction of administrative overheads through workflow. WSS 3.0 lists are required to store data and are used for workflow tasks. The list design is such that it is ubiquitous enough to use in any of the departmental site for the same purpose. The same re-usability is designed into the results lists used for role-based implementations. Each template is made by creating a list from pertinent site columns, and the newly created list is then saved as a template. The template is then added to the system and chosen for a departmental site. The results templates are modified to include custom columns inherent to a given testing activity.

Table 4.2, Table 4.3 and Table 4.4 shows the resulting list structure.

Table 4.2 Cardiology results template

Department	**Cardiology**								
List Title	Cardiology BTR								
List URL	http://community.stirlinghealthcare.com:81/cardiology/Lists/Cardiology BTR								
Columns	Date Tested	PatientID	WBC	RBC	As-signed to	Status	Modified By	Cre-ated By	Tested For
Types	Date time	Custom Site Column	Num-ber	Num-ber	Person	Work-flow	Default	De-fault	Person or Group

Table 4.3 Radiology results list design

Department	**Radiology**							
List Title	Radiology Results							
List URL	http://community.stirlinghealthcare.com:81/radiology/Lists/Radiology Results							
Columns	Date Tested	PatientID	Re-sults	Tested By	Status	Modified By	Created By	Tested For
Type	Date time	Custom Site Column	Im-age	Person	Work-flow	Default	Default	Parson or Group

Table 4.4 Generic appointment list template

Columns	**Title**	**Appointment time**	**PatientID**	**Assigned to**	**Location**	**Information about your appointment**	**Additional Notes**	**Created by**	**Modified by**	**Status**	**Appointment Type**
Type	Text Link	Custom Type and site column Date Time	PatientID or Person with image	Person	Location	Link	Multiple Lines of Rich Text	Default	Default	Workflow	Choice

4.7. Data Integration Model

Disparate data representing a NHS CRS or a GP EHR system is combined with native MOSS system data to form equally useable profile attributes and is used to meet the objectives of integration of disparate data in a useful way. The patient then views the complete profile record through their My Site interface. Workflow governs access for HCPs to the patient's site thus making the record available on a need-to-know basis similar to (Russello *et al.*, 2008). Figure 4.4 shows a basic overview of the import process.

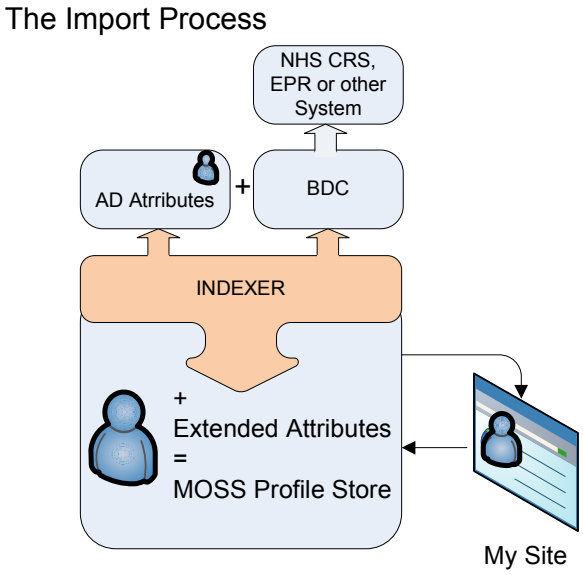


Figure 4.4 Data integration through profile import

4.8. HCP Role Based Security Model

The security design is based on a static hierarchical model based on mixed with a patient-centric changing, moving matrix that manipulates the static model. The overall security design is based on least privilege network security principles applied in a personalised context. The dynamic aspect of the design comes from consent-based workflow. As a patient moves around from department-to-department, HCPs are assigned. The patient is required, as part of the process, to accept or reject a given stage. Access to departmental list items is given to the patient to complete tasks such as accept an appointment. At the same time the assigned HCPs are given access to a patient's my site. At each stage in a patient's journey, a new HCP is added to the site this gives the patient a quick view of treatment history and allows them to look at the HCP's My Site to gain highly visible aspect of the people that are caring for them. At the end of a case, patients are removed from items , and HCPs are once again prevented from accessing a patient's site satisfying a need-to-know principle, as shown in Figure 4.5.

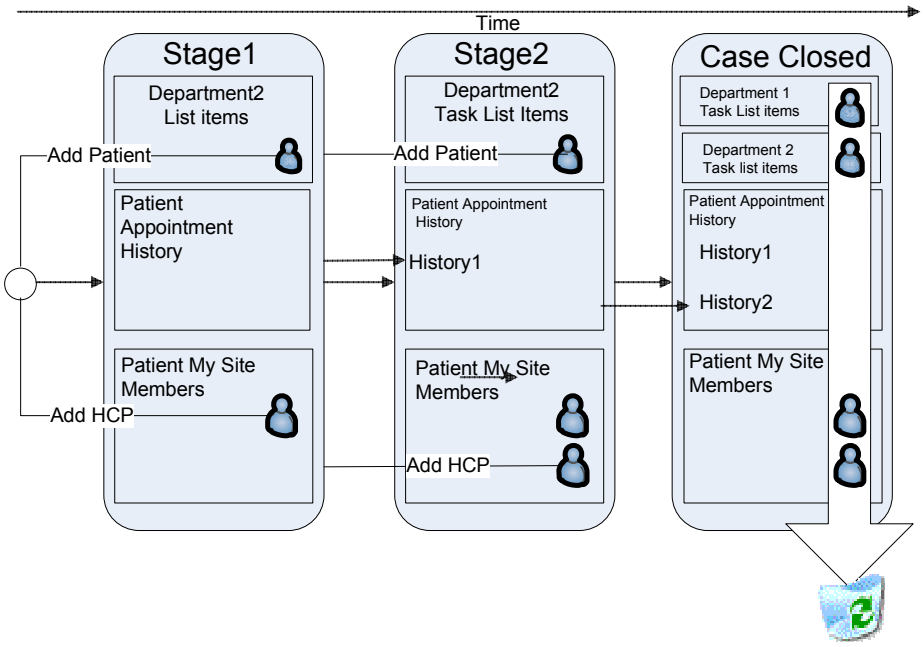


Figure 4.5 Dynamic security process

In the static model, groups are defined by organisational hierarchy. Objects relating to the hierarchy are stored in lists, in logical department locations. Access to lists is cross department therefore relevant permissions are assigned. Table 4.5 shows the designed access matrix. Personalisation is compulsory for the patient but flexible for the HCP, as patients cannot be expected to interact with the system at all times such as in an accident and emergency situation.

Table 4.5 List access matrix and permissions.

Group / List	Radiology Owners	Cardiology Owners	GP Administrators	GP Doctors	Surgery Surgeons	Surgery Administrator
Cardiac BTR		Contribute	Read	Read	Read	
Cardiac Appointments		Contribute	Read	Read		Read
Radiology Results	Contribute			Read	Read	
Radiology Appointments	Contribute		Read	Read		Read
Surgery Appointments			Read	Read	Contribute	Read
GP Appointments			Contribute	Contribute		
Surgery referrals			Contribute	Contribute		Contribute

4.9. Interface design

My Site features as the central interface for a patient. A patient interfaces with their My Site to gain information about their history or record. Patients will only see data relevant to themselves. Also, patients can see, at a glance, who their current HCPs are and, using the security model, HCP only have access to a patient's My Site when they involved in a case. Implemented elements are prefixed with the "My". Figure 4.6 is a high-level combined representation of both pages in a patient My Site interface. One important aspect of this design is the inclusion of a photographic image for both patient and HCP. A patient may wish to see what their HCP looks like by visiting a HCP my site and a HCP can use a patient's image to reduce error or prevent fraudulent medical care claims.

HCP use My Site in much the same way except the member of patients and other HCPs to a HCP's My Site is constant and not affected by workflow. This increases visibility across the organisation as HCPs can search for other HCPs based on knowledge of such attributes as skill sets, and so on. Patients can then access department data via personal links (Wodtke, 2003) that lead to personalised views of current case status including test results and appointments. These personalised views consist of web part pages containing page element web parts that display filtered views of the list architecture based on the person accessing the system. The high-level design of this page in shown in Figure 4.7**.**

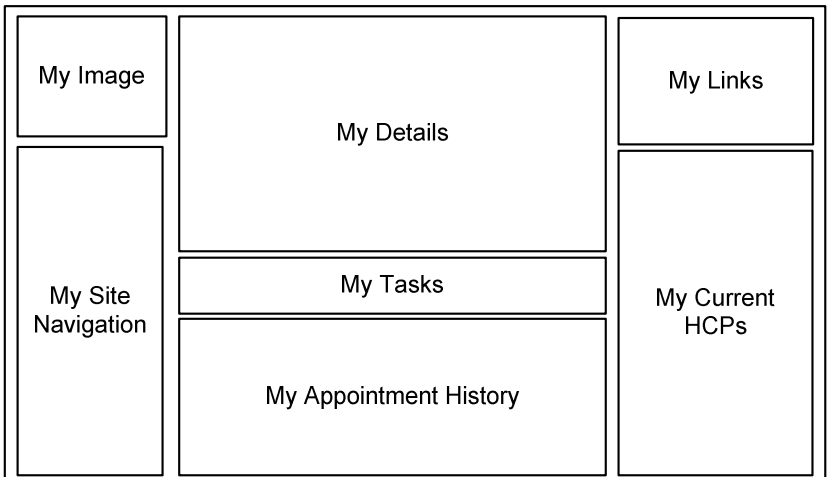


Figure 4.6 Patient interface design

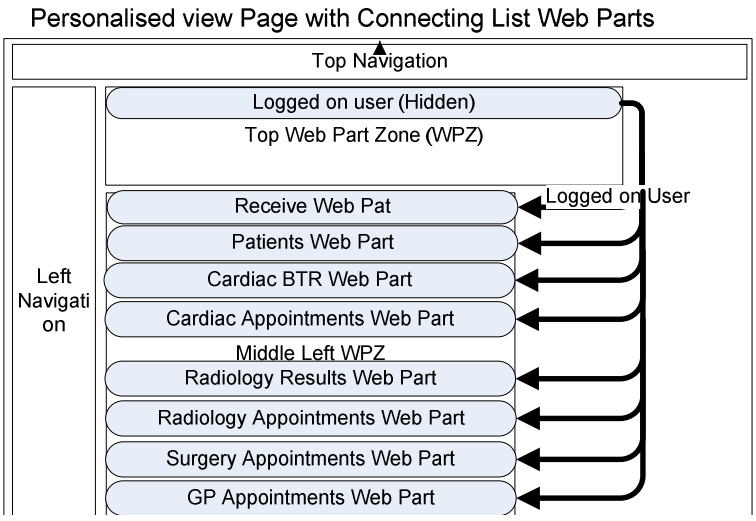


Figure 4.7 Personalised view page design

4.10. Information Targeting Design

Information on various systems and lists are general to the department. Without targeting of relevant information, the user, patient or HCP must drill down into this data to acquire relevance; however, law and policy prevent patient and HCP from seeing information about other patients. One way to cope with this is to copy data into an area that has access; the consequence of which is data repetition. Providing views of lists and disparate sources, targeted at individuals, through a single interface reduces complexity to the user, integrates data, increases visibility and extends reach. The coloured dot clusters in Figure 4.8 represents data sources. Each colour represents a patient or a HCP. Figure 4.9 shows the effect of adding targeting to the issue depicted in Figure 4.8. This design is based on the interface and security models described in Sections 4.8 and 4.9. Information is filtered at a personal level on patient's My Site. On each department site, personalised views of lists such as appointments views are accessible via web part pages (the same model can be applied to HCPs). The personalised views mentioned in Section 4.9 are where this design is applied, and are secured by making default views person-centric and preventing access to any other view for patients and HCP in combination with the static security model described in Section 4.8.

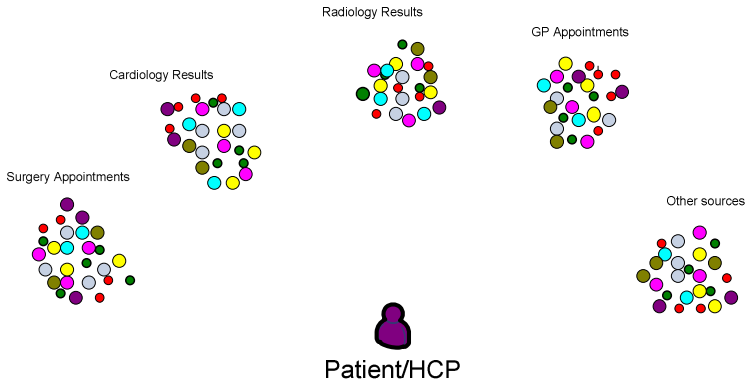


Figure 4.8 The issue of unfocussed data

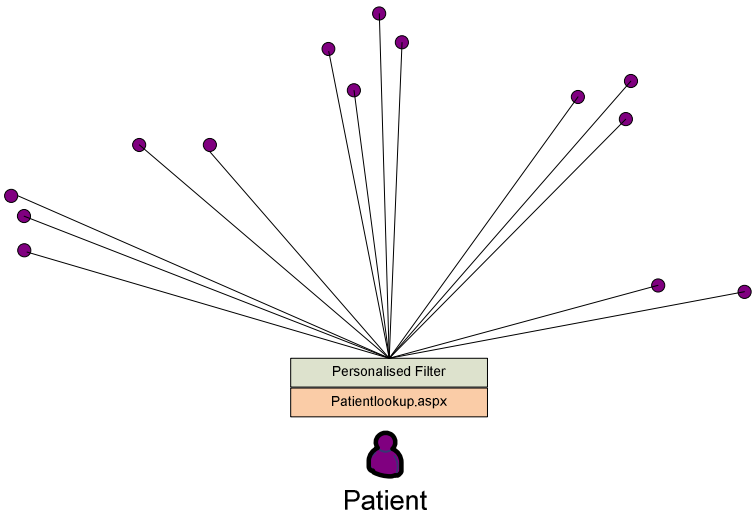


Figure 4.9 Effect of personalised filtering

4.11. Workflow Design

This workflow design meets the objectives of reducing administrative overheads by cutting out paper-based operations from patient management. In addition, this design reduces the risk of error by removing a manual processes like the postal system from the process. This design also allows for extraction of management information from day-to-day tasks (NHS Management Executive, 1992) by recording a completed workflow in logs. Finite state machine design methodology is used as a basis for creating a consent-based, SharePoint workflow based on the following, basic scenario: "A patient is referred for minor surgery by a HCP".

4.11.1. Business Process Analysis

Upon analysis, there are several paths the information must take and a number of negotiations must take place for the patient to know where and when they are supposed to be. When a patient is referred, the referral is done by the doctor either electronically or by mail. The ap-

pointment is received, assigned to a HCP based on availability and, if accepted, a letter is sent out to the patient with a suggested date and time, the letter may contain:

- Information about the hospital location, map and parking Instructions.
- Information on the procedure.
- Department location.
- Special instruction such as sample requirements or procedural dependencies.
- Contact information.
- Appointment time.

At this point, the patient may reject the given time requesting a new appointment, in which case the process between patient, surgery administrators and surgeons will have to begin again. If this is the case an additional letter is sent containing the new appointment time. Once a patient has accepted the appointment, it is then expected to have taken place by a given period after the start of the appointment. When the appointment has either taken place, or has been terminated, it can then be regarded as ended. This scenario can be represent by a flow diagram as shown in Figure 4.10.

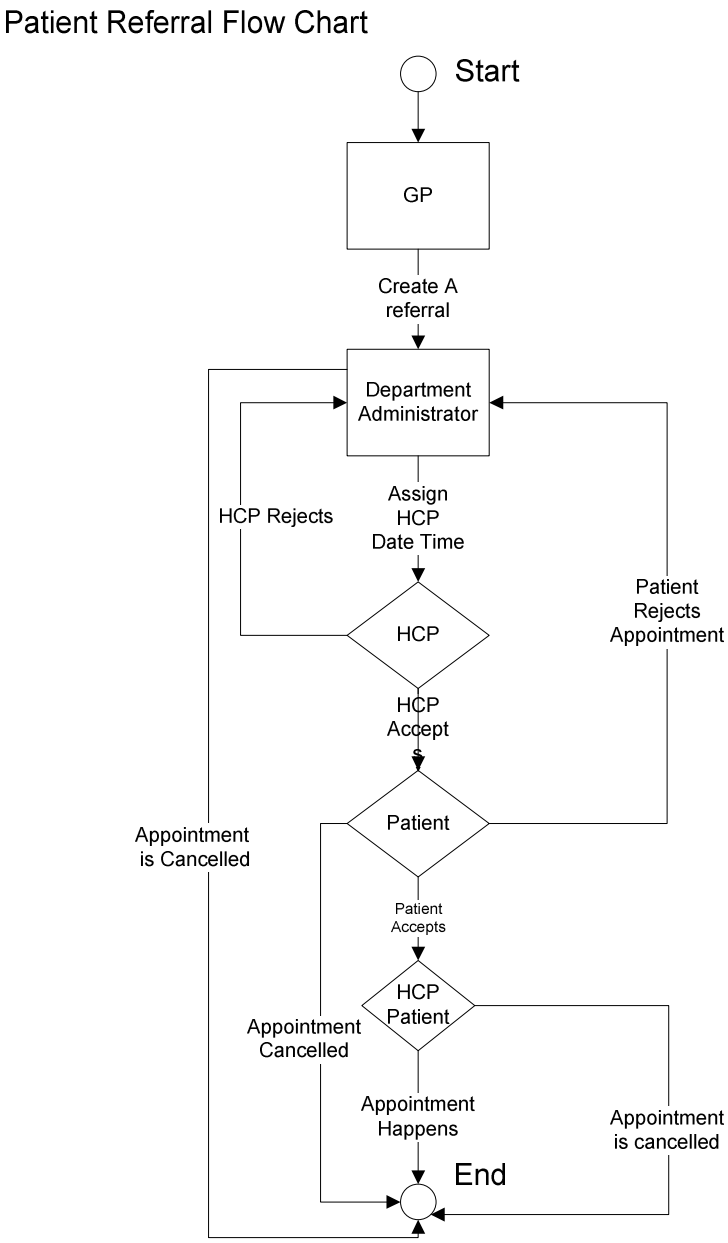


Figure 4.10 Flow diagram of business process

4.11.2. Finite State Machine (FSM) Model

Another way to describe the scenario would be in terms of state; in a state machine approach an appointment will have states (Katz, 1993). When analysing the scenario the following states can be drawn from the text:

- The process does not exist.
- The process is started and waiting for department activity.
- The process is waiting for HCP activity.

- The process is waiting for patient activity.
- The process does not exist.

In addition to state, there must be events to invoke actions that cause state to change. From the description of the business process the following events can be drawn (Thuemmler, 2008):

- A GP or other HCP initiates a referral.
- Waiting for HCP: The department administrator to which the referral has been assigned assigns a HCP giving a date time and location.
- HCP Accept: An HCP accepts the appointment time given by the department administrator and information is sent to the patient.
- HCP Reject: A HCP rejects the appointment time.
- Patient Accept: A patient accepts a suggested appointment time.
- Patient Reject: A patient rejects the suggested time, or for some other reason.
- Cancelled: There is a cancellation either by the Department, the Patient or the HCP.

The actions in FSM approaches defines what the system must do to satisfy a state transition. Actions here are thought of in terms of required MOSS activities such as send an email or update a task:

- Send an email.
- Update task.
- Create task.
- Assign task.
- Create a log entry.
- Add a user.
- Add a list.
- Add to a list.
- Update a list.
- Remove a user.
- End task.

Using finite state machine methodology, the business process can seen as a state transition diagram shown in Figure 4.11.

The .NET 3.0 development framework supports both sequential and state-based workflow and it was decided that a state-based workflow would be used in this scenario. The state based approach seemed to suit the business process and realising the design from the scenario in this way was more methodical. In addition, it would beneficial to ascertain the degree to which established state based theory can be applied in MOSS.

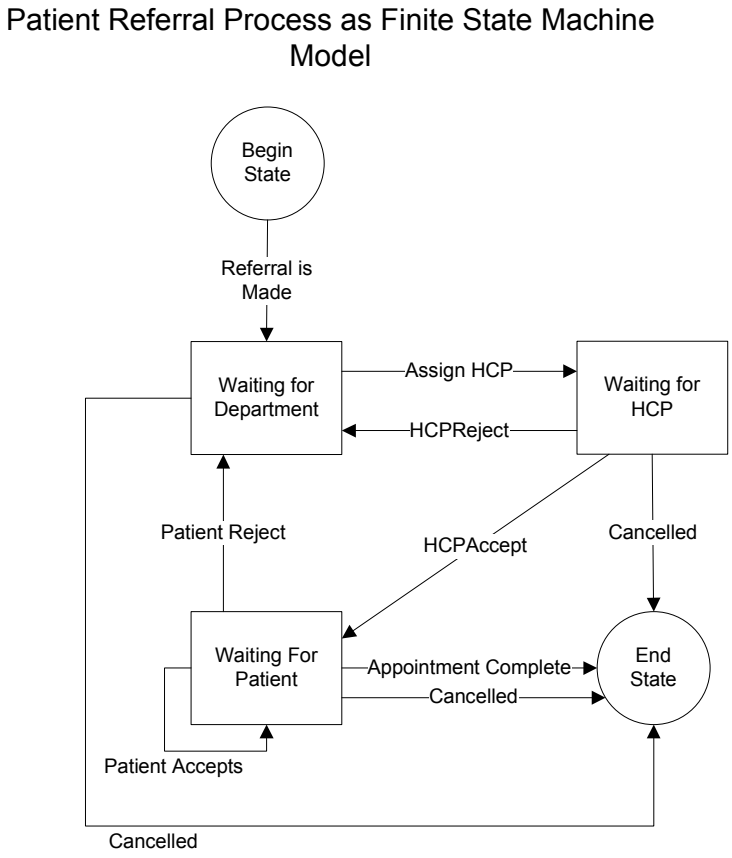


Figure 4.11State transition diagram modelled on workflow scenario

4.12. Conclusions

This system comprises of three approaches, the aspects of which when combined, are designed to meet the objectives of reducing complexity, increasing visibility and relevance-through-role, and reducing administrative overheads. Implementing the data integration design allows patient information to comprise of many different sources. Realising security models and targeting information could increase relevance through role and implementing workflow may decrease paperwork and remove administrative function through automation. This inclusion of the patient as a user of the system, combined with the gathering from information from different department and disparate data sources, supports the patient-centric approach as defined by (IBM, 2006) and (Department of Health, 2000). The overall technical approach to this design is based on the theoretical capabilities of MOSS and WSS observed through analysis and literature review.

5. Implementation

5.1. Introduction

This chapter described the steps used accomplish the implementation of the designs outlined in Chapter 4. It begins with methods used to implement the core enterprise structure required to support MOSS. This is followed by description of the implementation of HCP department WSS site structure and WSS list templates. The chapter concludes by detailing of the methods used for implementing combined patient records with data integration, targeting information based on role and workflow.

5.2. Infrastructure Implementation

Each server role represented a required component or dependency of the main MOSS framework. To accomplish the creation of the enterprise environment required, virtual server technology was used (a short discussion of virtual server technology can be found in Appendix J). Each virtual server was setup by installing Microsoft Windows Server 2003, IIS, ASP.NET, and .NET Framework 2.0/3.0/3.5. Server roles such for Microsoft Active Directory (AD) and Domain Name Service (DNS) were assigned by configuring the server operating system to produce a network domain (Minasi, 2003). Microsoft Exchange Server 2003 (MSE2003) was then installed on one server. Microsoft SQL Server 2005 (MSSQL2005) on a different server and MOSS on the remainder. The basic network topology of the enterprise was configured as is illustrated in Figure 4.2. Table 5.1 shows the roles assigned and technologies installed onto the basic operating system.

A Top Level Site (TLS) collection was created on SASSPSWFE1 server using the SharePoint administrative interface. Four Windows SharePoint Services 3.0 (WSS 3.0) team sites "GP", "Surgery", "Cardiology" and "Radiology", were created underneath the top level site (see Figure 4.3). A Shared Service Provider (SSP), required to host My Sites, was then created and hosted within the TLS structure and administered from the address "http://comminuty.stirlinghealthcare.com:81/ssp/admin".

Table 5.1 Server configuration

Server Name	IP	Role	Technology
Crl-wsgdev-2	192.168.1.65	Directory Server Email Server Domain Name Service	AD MSE2003 DNS
SASSQLDEV01	192.168.1.68	Database	MSSQL2005
SASSPSWFE1	192.168.1.67	Web server	MOSS, IIS, ASP.NET,.NET 2/3/3.5
SASSPSSearch	192.168.1.71	Index and Search	MOSS, IIS, ASP.NET,.NET 2/3/3.5

5.3. Test Users

14 test users, representing HCPs, were added to Active Directory (AD). The HCP users were divided by role and department representing the administrative and practical fields of medicine, such as Doctors and department administrators such as receptionists, or personal secre-

taries. Each HCP's first and second name, department name and location were added to the first name, second name, department and office attributes in AD, respectively (See Appendix A, Table A 1and Table A 2). In addition, the fields representing Job title in AD were populated with a username as they also represented the HCP's job title. Appendix Table A 1 shows how Appendix Table A 2 above is represented in AD, and three test patient user accounts were created in AD also found on Appendix Table A 2.

5.4. Disparate data consumption

Patients are added to the system as users of the system. Supplementary data, representing other attributes of a patient's personal information is then consumed by the system and used to build an extended profile of each patient; each profile consisting of both native user directory attributes and attributes made up from the supplementary source. The profile data is then expressed on the patient's own My Site, as described in Section 4.7.

5.4.1. Single Sign-On (SSO) Configuration

The PostGreSQL database described in Section 5.4.2 uses a different authentication method to Microsoft's integrated authentication. In addition, only one user ("matt") was configured to access the provided database instance. To allow BDC access using these credentials, MOSS's SSO (Appendix) service was implemented, SSO is a service comprising of a MOSS database storing encrypted remote database and web service credentials, and a mechanism, known as "Privileged Group", by which AD security groups can be assigned to the stored credential. The BDC makes a call to the SSO provider when a BDC method is invoked.

5.4.2. Disparate Data Source Configuration

In configuring the disparate data source, a database entitled "SCU_Portal", representing a NHS CRS, EPR, ESR or Spine (See Section 2.5.3), was created on the PostGreSQL database instance located remotely in relation to the project servers. A table entitled "profilesupp" was created on the SCU_Portal database, and columns were added to the table and data was manually entered (see Table 5.2). The PID column in the profilesupp table contains the same data as the username attribute in AD, and the additional fields contained other data obtained from healthcare researchers (Thuemmler, 2007), such as blood type, sex, ethnic origin and current GP, which represent a sample of the data that meets the following conditions:

1. The attribute does not exist in AD.
2. The attribute does exist as Structured Query Language (SQL) compatible data in another health care system

Table 5.2 Disparate database table

PID	BLDTYP	ETHNO	SX	CGP
Patient01	O+	1	F	THOMSON
Patient02	AB-	3	M	BLACK
Patient03	O	1	F	THOMSON

5.4.3. Account Setup

Patients Patient01, Patient02 and Patient03 were added to AD as new user objects. Each patient's Forename Name, Family Name (See Appendix A1 and A2), Username and Password were added to the corresponding attribute in AD, all other attributes were not required for this implementation so they were ignored. Each patient was provided with an email box on the provided domain email server in the format of <forename_initial.Family_name@stirlinghealthcare.com>. The SSO service was started, username details and a password were added to the SSO database and the privileged group was set "OFFICE\Domain admins" as the OFFICE\administrator user is used for testing purposes throughout the system.

5.4.4. BDC Application Definition File

Using an eXtensible Markup Language (XML) editor, an Application Definition File (ADF) is created using the hierarchical node representation described in Section 3.6. The commented mark-up can be viewed in the Appendix; Section A.2 The ADF file was loaded into the BDC for community.stirlinghealthcare.com:81. Verification that the BDC entity was working was carried by using a test URL that invokes the SpecificFinder (See Section A.2) method of the BDC entity.

5.4.5. Configuring Profile Import Connections

Profile import is based on a master connection, usually to AD; supplementary connections can then be added which can comprise of alternate sources one of which being the BDC. A supplementary connection was added to the master import connection and configured to use the BDC in a 1:Many relationship based on the "Username" attribute in AD and the "pid" column in the profilesupp table.

5.4.6. Initial Testing, outcomes and Error Correction

A profile import was executed and the logs examined to verify a successful import. The first attempt threw an error describing the requirement of a finder method. At this stage the ADF file only contained a specific finder method; a Finder method was thus added to the ADF see Appendix A.2. The import was tested again and inspection of the logs appeared to indicate a successful import. Closer inspection of the logs revealed that the Indexer engine is responsible for populating the Profile Store as the crawl log showed Lightweight Directory Access Protocol (LDAP) strings in the queries it was making. In addition to these queries the crawl URL "$$$nonmaster$$$" was present in the log indicating the attempt to crawl the BDC supplementary profile connection. Unfortunately there were errors associated with this entry. Inspecting the trace logs from the Indexer revealed a problem with the connection to the BDC. It was realised that the index server did not have the ODBC drivers installed on it and could not connect to the data source. The ODBC drivers for the PostGreSQL database were then installed on the server configured to be the index and a successful import was recorded.

5.4.7. Results and Verification

After a successful import, new user attributes are created and mapped to the BDC columns, the new attributes are configured to display in a user's profile properties on their respective

My Site with an identifiable section titled “Custom Properties”, an additional profile import is then executed and verified. Each patient user account is then used to access the system followed by visiting the respective My Site to see the results. An example of one patient user’s My Site is shown in Figure 5.1.

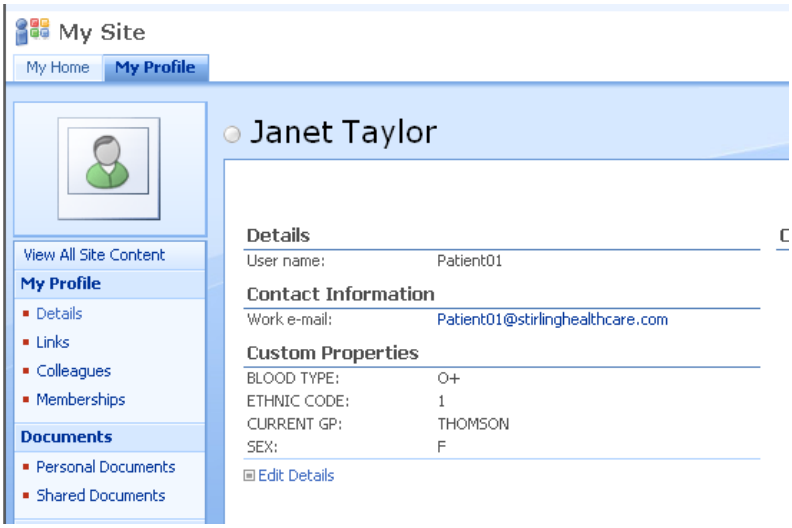


Figure 5.1 Screen shot showing disparate in "Custom Properties"

5.5. Information Targeting Implementation

Several SharePoint lists are created using templates, such has information to imply datasets such as blood test results, X-rays, cardiographs, appointments and personal details. The created lists are then added to each respective departmental site. Patient and HCP My Site are then configured to display personalised view of data.

5.5.1. List Configuration

List templates are created using the designs from Section 4.6, such as results lists in Cardiology and Radiology from templates and populated with pertinent data. X-ray images obtained from Radiology Society North America are then added to the default list of type picture library (Appendix B.2) for the Radiology WSS 3.0 site. Links to the pictures are used to populate the radiology results list. Cardiology results are made up with data representing Red Blood cell Count (RBC) and White Blood cell Count (WBC) using information obtained from (Leeds & Bradford Pathology Partnership, 2007). Appointment lists, created from the designed templates, are added to each department site and renamed appropriately. See Appendix E for details of lists and content.

5.5.2. Security Configuration

Site groups are created using the Group Names and memberships shown in Table 5.3. Each department site list is then configured using the security access matrix described in Section 4.8. In addition the default public list view is configured to be filtered based on the person looking at the list. Additional views are then created for HCPs department applications.

Table 5.3 Sharepoint groups

Site Group Name	Members Username
GP Administrators	OFFICE\GPSecretary
GP Doctors	OFFICE\GP
Surgery Administrators	OFFICE\SurgeonSecretary
Surgery Surgeons	OFFICE\Surgeon
Radiology Owners	OFFICE\RadioLab
Cardiology Owners	OFFICE\CardioLab

5.5.3. Interface Configuration

An aspx page "patientlookup.aspx" is created below the TLS. At this point, a limitation was discovered in the system which meant to achieve the desired result; a deviation from the original design was required. Content Query Web Parts (CQWP) has thus been added to the page, one for each list created in Section 5.5.1. Each web part is then configured to look at each list, altered it using XML and eXtensible Stylesheet Language (XSL) and filtered to the person looking at the page. The method used to filter contents differs slightly to that discussed in Section 4.9 as connections could not be used in a CQWP. A link to this page is then placed in the patient's My Site in the My Links section discussed in Section 4.9.

5.5.4. Testing

To ensure the security matrix is implemented correctly, access rights are tested by logging as a particular user and trying to perform a prohibit function using the access matrix described in Section 4.8. To ensure the correct list data was displaying on the patientlookup.aspx page, comparisons are made with the screen output and the original list contents located in Appendix C. Figure 5.2 shows the patientlookup.aspx page as seen by Patient01, and Figure 5.3 shows the same page as seen by the Surgeon user.

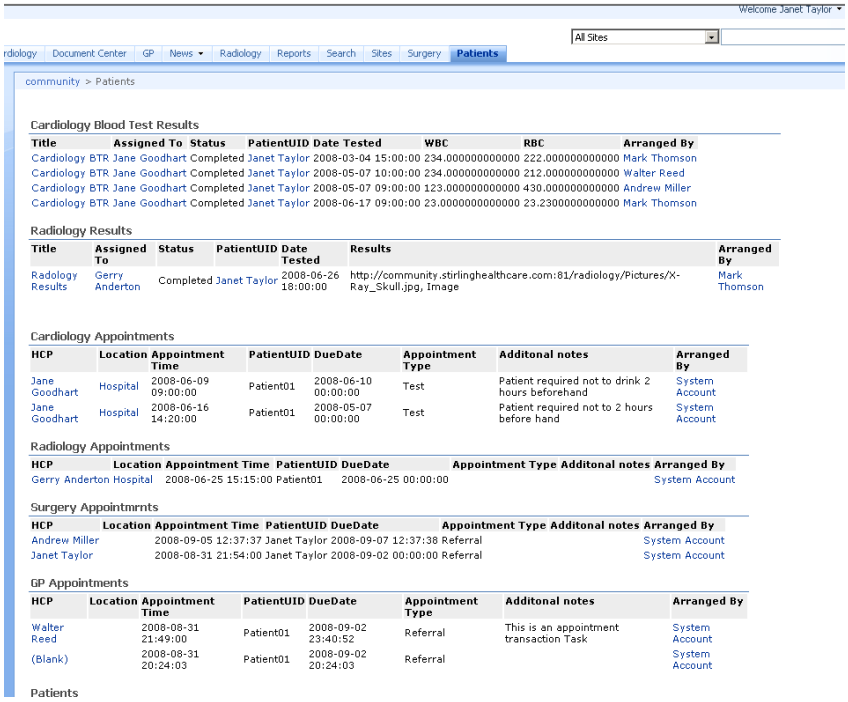


Figure 5.2 Information targeting interface as seen by Patient01

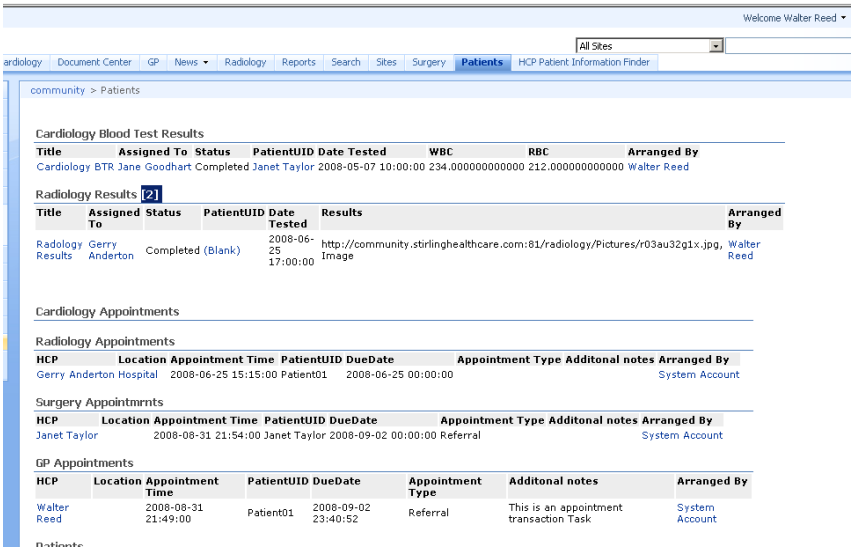


Figure 5.3 Information targeting interface as seen by Surgeon user

5.6. Workflow Implementation

Visual Studio 2008 (VS2008) is used to create a custom SharePoint state machine workflow based on simple analysis of a hypothetical business process expressed as a FSM Model. The first requirement for this implementation is to translate the FSM design into a SharePoint workflow terminology.

5.6.1. Tasks and Task Lists

To support the referral scenario described in Section 4.11, some of the task lists (appointments) created in Section 5.5.1 are modified to include columns required for workflow functionality. The task lists consist of special "Task" content types designed to work directly with workflow. Also the "Task" content type contains site columns which the workflow uses to function. Some of these site columns are:

- Due Date – the date or date and time a task should completed.
- Assigned To – the user the task is assigned to.
- Start Date – The date at which the task was originally created.
- End Date – The date at which the task status is set to complete or that the workflow is completed, programmatically.
- Status – The status of the task. This can either be direct representation of the state in a state-based workflow, or a more user-friendly representation of a given business process.

When using workflows within SharePoint, a task list is used to assign individuals with activities that surround a list item known as a workflow item. The workflow item exists on different list to task, which allows the task list to be used for both assigning multiple tasks facilitating parallel business processes and multiple workflow items.

In this method, a custom List named "Surgery Referrals" is created, inside the Surgery department site, to contain the workflow item. The "Surgery Appointments" List (Appendix Table E.1) created in 5.5.1 is used as workflow tasks list. The "Task" content type is then modified to accommodate such site columns as "Appointment time", "PatientUID", "HCPUID" and "Additional Notes". Usually, a custom content type is created containing these columns as modifying the base "Task" content type means the added columns would appear all workflows; this may be undesirable for alternative workflow applications.

In addition to the lists defined here, two more lists are used, these are created automatically by the workflow when an appointment is either completed or cancelled. "Appointment history" is created on the patient My Site and again for the department.

5.6.2. Business Process Activities to SharePoint workflow activities

To successfully implement the workflow, activities pertaining to each event is defined and mapped to MOSS workflow activities. In this case the patient is thought of as a *temporary* user of the surgery site appointment item as defined in Section 4.8. In terms of the framework setup so far, the patients and HCPs are governed by SharePoint lists so it reasonable to assume that any workflow would revolve around the manipulation of those lists. Table 5.4 shows the scenario events in terms of MOSS workflow activities.

Table 5.4 FSM events and MOSS workflow activities

FSM Event	Workflow Activities
A referral is made	When a new appointment is added to the Surgery Referrals List, the workflow is activated and some data such as additional notes is copied into the task list, a task is created and assigned to the department administrators group (in this instance this is a single user) and workflow enters the "waitForDepartment" state.
Department Assigns HCP	The Department Administrator or Personal Assistant (SurgeonSecretary sees tasks either via email or their My Site, The SurgeonSecretary edits the task, adding-in a time. They then add other information about the appointment such as links to appropriate documentation and location, and so on. They then set the task item state to "Waiting for HCP" and the workflow state transfers to "waitingForHCP"
HCP Accepts Task	HCP accepts the task and the status is changed to HCP Accept ; the task is then assigned to the patient, an email is sent, and the patient is added to the task list item in the security model to allow edit type access (See Section 4.8); the workflow process is set to the "waitForAppointment" state.
Patient Accepts	Patient sees the task in their My Site or via email and accepts the appointment time listed in the task by selecting Patient Accept in the task list status column. The workflow status remains on "waitForAppointment" state. This allows the patient to cancel at any point.
Patient Rejects	Patient rejects the appointment, the task is then assigned back to the department administrator and the process is started again the date and time are left to remind the administrator of what not to choose for the next attempt. The workflow process enters the "waitForDepartment" state and the operative is informed via email or by visiting their "My Site".
HCP Rejects	The HCP is not available for the suggested appointment, an outright rejection or reply, with a suggested new time, is sent back to the department administrators group and the workflow status is set to "waitForDepartment".
Cancellation	The workflow is terminated, and the status of the appointment list is set to Cancelled. A history of the cancelled appoint is logged in the patients list and a department appointment history list; The workflow is deactivated at this point.
Complete	The HCP accesses the appointment task and sets it to "complete". A flattened copy of the task item is added to both patient and department appointment history list providing logging and the task is removed.

5.6.3. SharePoint Workflow Activities

The programmatic equivalent of business process activities in SharePoint are known as SharePoint workflow activities. SharePoint includes its own workflow activities designed to work with the tasks list mentioned in Section 5.6.1. Understanding these activities and how they map to a business process are necessary to implement a successful workflow. The SharePoint workflow activities used in this method are:

- Code – allows the insertion of any piece of code in the language the workflow project is written in. This can be any .NET Common Language Runtime (CLR) compatible language (Walther, 2006).
- onWorkflowActivated – This activity instantiates the workflow and is responsible for providing a Globally Unique Identifier (GUID) and setting the workflow status column in the workflow item to "In Progress".
- setStateActivity – This activity moves the state machine workflow into a predefined state.
- createTask – This activity creates a task in the workflow, and the created task is added to the task list for the workflow. Properties of the createTask activity include the fields in the "Task" content type described in Section 3.4. A GUID is also created allowing several tasks to exist on the same list.
- onTaskChange – This activity listens to the task and triggers when anything on the task list item is modified, it has before and after properties allow for tracking, audit and roll back.
- upateTask – This activity allows a task to altered without triggering the onTaskChange event
- IfElse – The ifElse branch activity is the basis for predicates in the proposed state machine workflow, however it also acts as a programmatic logic branch in any workflow.
- Conditioned Activity Group (CAG) – This activity allows several activity sequences to run at once based on the condition of another aspect of a workflow, and has been referred as "Multi Threading" workflow, however, its operation does not reflect the mechanisms by which the term is usually associated (Shelton, 2007).
- Delay – A delay effectively makes the workflow wake up after a given time interval, and to save processing resources a workflow is only considered to be awake at the point at which something is changed and immediately after. Beyond this point the workflow sleeps by freeing up the resources it was using. This is acceptable, but cannot provide timing constraints. This means that a task attribute such "Due Date" cannot be monitored and could easily pass while the workflow is dormant. Adding a delay activity in conjunction with another activity rectifies such issues.
- stateInitialization – This activity is fired when a state is entered, other activities can be placed inside this activity if the functionality is required.
- eventDrivenActivity – This activity hosts all the child activities that manipulate the workflow. For example the onTaskChanged activity will not work in a state machine workflow unless hosted by this activity.

- taskComplete – This activity updates the workflow task item to complete entering information into the SharePoint database.
- terminate – This activities terminates the workflow itself.

5.6.4. Workflow Activation

A workflow is activated in SharePoint either: manually; automatically by the addition of a list item; or when an item is changed after it has been added to a list. In this case the workflow is activated automatically by the addition of a new appointment referral to the "Surgery Referrals" list.

5.6.5. Events and Predicates

As defined in finite-state machine literature, events trigger actions that cause state transition and predicates are used for transitional dependencies, that is, the predicate condition determines the action which causes the transition. Predicates are usually a Boolean, true or false data type. In the case of the activities listed in Section 5.6.3, there are several predicates and one event. The event that triggers actions that causes any state transition in this scenario is any modification of the task list item. In a production environment, the form used to allow a user to edit a task would only display the elements intended for editing. In this method, timing constraints prevented forms from being created as it required research into another WSS client technology (Janus, 2007). Therefore an incorrect modification in some states could cause an error. Predicates determine what task list item status has changed to and consequently which state to move to. Figure 5.4 shows a code implementation of a .NET 3.0 workflow predicate written in C#. This particular predicate checks the task status field to see if the patient has rejected an appointment.

```
// Predicate, asses the status of the workflow to establish what action to take. onPatientChange is the event this predicate is
 used for. It it used when a change is made to the task in the waitingForAppointment state

    private void rejectOrCancel(object sender, ConditionalEventArgs e)
    {
        Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
        string status = onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString();

        if (onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Patient Reject"))
        {
            e.Result = true;
        }
        else
        {
            e.Result = false;
        }
    }
```

Figure 5.4 Predicate expressed in .NET 3.0 workflow

Using the workflow activities discussed in Section 5.6.3 and a module of Visual Studio 2008 called the Workflow Designer which maps out empty code method elements of the workflow class structure using a visual design interface; the work flow is laid out starting with the states discussed in Section 4.11.2 and illustrated in Figure 4.11. These states were:

- Begin state – Workflow1InitialState see Figure 4.11.

- Set Task – (See Section 5.6.6.)
- waitForDepartment.
- waitingForHCP.
- waitForAppointment.
- End State – testState.

5.6.6. SetTask State

In addition to these states, the programmatic constraints of the .NET 3.0 framework require the implementation of an extra state that creates the initial workflow task item for the SurgeonSecretary user, this state is titled “Set Task”. Figure 5.5 shows these states.

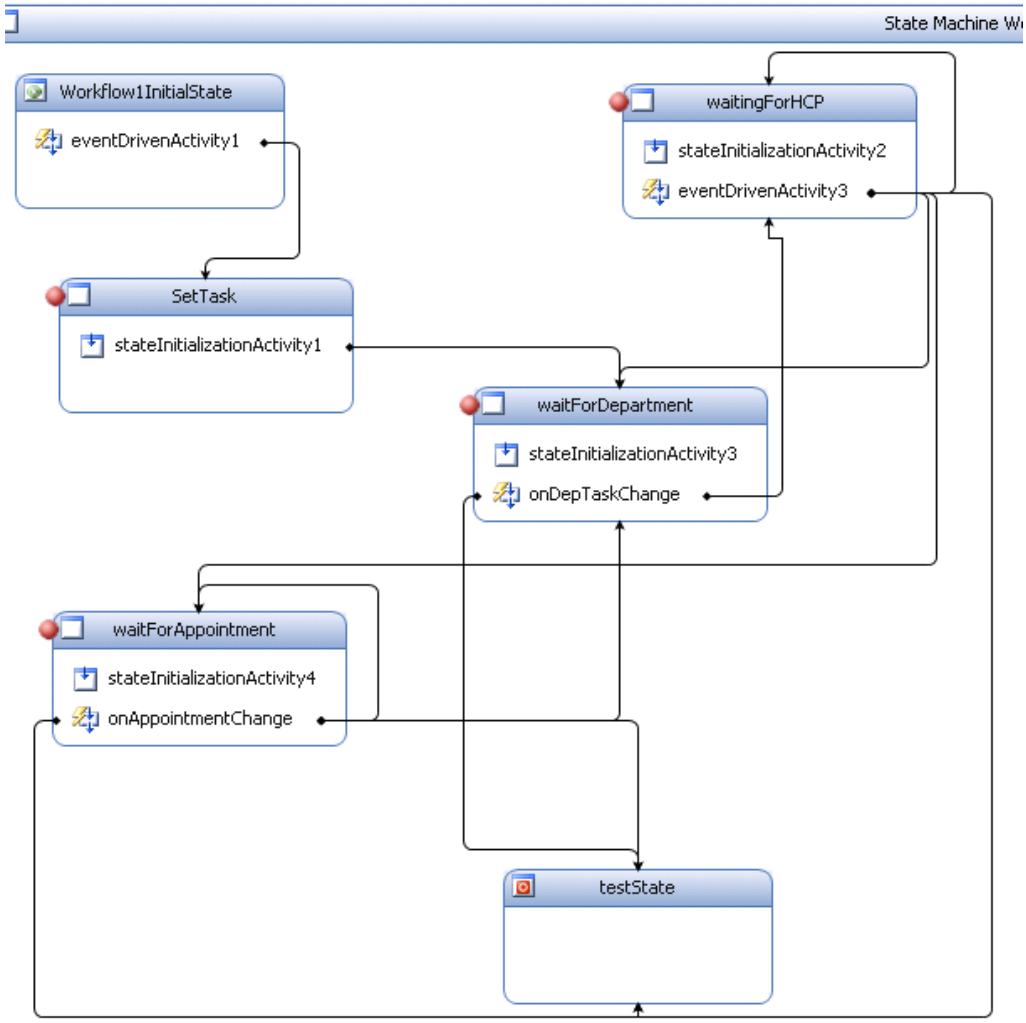


Figure 5.5 Screen shot of development design surface showing SetTask state

5.6.7. Code Class

A full code listing for this is available in Appendix H. Using the C# programming language, the code method containers required to implement the workflow are added by placing activities into the workflow designer. The methods are then completed by adding the appropriate code to the created method containers. Figure 5.6 shows a screen shot of the "Create Task" activity in the Visual Studio 2008 workflow designer. Figure 5.7 shows the completed code method for the createTask1 activity shown in Figure 5.6, while Figure 5.8 shows the completed code method for sendInitialEmail activity shown in Figure 5.6. The setStateActivity4 activity shown in Figure 5.6 is responsible for the state transition to the waitingForDepartment state; no code is required for this activity.

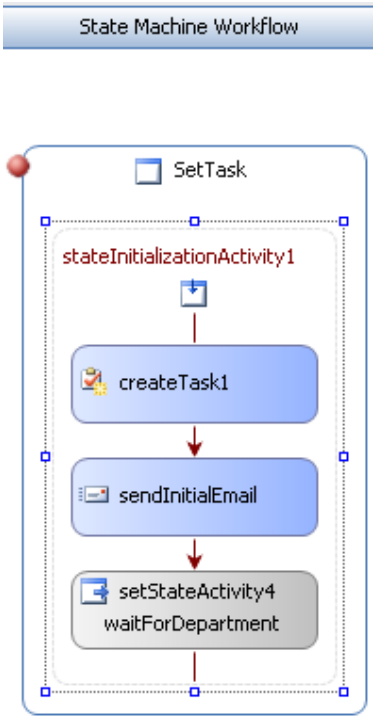


Figure 5.6 Workflow activities in the set task state initialisation

```
//When the setTask state is reached, the stateInitialiasation activity runs this methods to set the task. The empty method is created by dragging a Create Task activity onto the VS2008 design surface
    private void createTask1_MethodInvoking(object sender, EventArgs e)
    {
        SPListItem currentItem = workflowProperties.Item;
        char delim = ('#');
        string s = currentItem["PatientUID"].ToString();
        string t = currentItem["Created By"].ToString();
        string[] created = t.Split(delim);
        string[] words = s.Split(delim);


        createTask1.TaskId = Guid.NewGuid();
        //createTask1.TaskProperties.AssignedTo = @"office\SurgeonSecretary";
        createTask1.TaskProperties.Title = "New appointment referrral";
        createTask1.TaskProperties.Description = "This is an appointment transaction Task";
        createTask1.TaskProperties.DueDate = DateTime.Now.AddDays(2);


        createTask1.TaskProperties.ExtendedProperties["PatientUID"] = @"office\" + words[1];
        createTask1.TaskProperties.ExtendedProperties["Appointment time"] = System.DateTime.Now;
        createTask1.TaskProperties.ExtendedProperties["Appointment Type"] = currentItem["Appointment Type"].ToString();

        createTask1.TaskProperties.HasCustomEmailBody = false;

        createTask1.TaskProperties.SendEmailNotification = false;
    }
```

Figure 5.7 Create task activity implemented method

```
// This email activity sends the email to the SurgeonSecretary user alerting them of a new task. The empty method is created by dragging an email activity on the VS2008 design surface
    private void sendInitialEmail_MethodInvoking(object sender, EventArgs e)
    {


        SPListItem currentItem = workflowProperties.Item;
        char delim = ('#');

        string s = currentItem["PatientUID"].ToString();
        string u = currentItem["Created By"].ToString();
        //string[] ass = t.Split(delim);
        string[] created = u.Split(delim);
        string[] words = s.Split(delim);

        this.sendEmail1_Body1 = "<p>" + words[1] +
            " has been reffred by " + created[1] + " on " +
          System.DateTime.Now.ToString() + ".</p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) + " and add an appointment time before the due date of " +
            System.DateTime.Now.AddDays(2) + "</p><p>Thank you</p>";
        this.sendEmail1_Subject1 = "New Appointment Referral";
        this.sendEmail1_To1 = "surgeonsecretary@stirlinghealthcare.com";
    }
```

Figure 5.8 Send email activity implemented code

5.6.8. Verification

The workflow is deployed to the “Surgery Referrals” list and set to trigger when an item is created. Using the “GP” user account, an item is added to the surgery referrals list simulating a HCP referring a patient and the “PatientUID” field is populated, and the list item is then submitted. Using the Visual Studio debugging feature, the workflow progress was followed from a programmatic context as it set the task and sent an email. The “Surgery Appointments” list is then checked to verify a new task item was created by using the account to which the task was assigned. Figure 5.9 shows all the tasks assigned Andrew Miller (Sur-

geonSecreatary) as the Surgery Appointments List is configured to display personalised data, see Section 5.5.2.

Figure 5.9 Screen shot of appointment list after item added to referrals list

The user "Andrew Miller" is then used, and his My Site is inspected using the account to verify that the new task could be seen. This is shown in Figure 5.10. The user's My Site shows all the tasks assigned to the user, and Andrew Miller's mail box is then checked to verify the existence of mail informing him of the new task. This is shown in Figure 5.11.

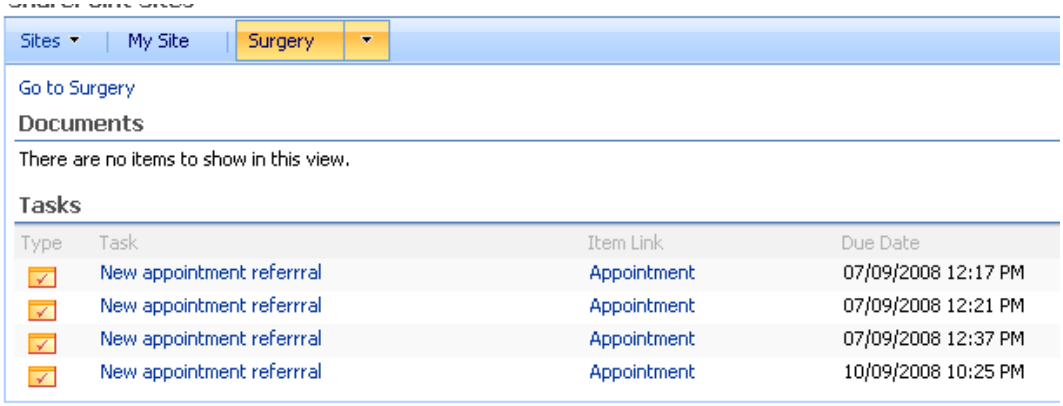


Figure 5.10 Screen shot of SurgeonSecretary user's My Site showing tasks

Figure 5.11 Screen of email sent when task is assigned

The task is edited using the “Andrew Miller” user account, the appointment time is then edited simulating the surgery administrator suggesting an appointment time, then assign assigned to the user “Walter Reed”, the Surgeon user. The task is then edited using the “office\SurgeonSecretary” user account of Andrew Miller. A verification is also made that the task appeared in the tasks list present in the “My Site” for the “Office\SurgeonSecretary” user. There is also a verification that the tasks list is accessible from a mobile device. The task is then accessed and edited including the “Appointment Time” field. The task is then set to “Waiting For HCP”. Upon submission of this change to the task; the debugger in Visual Studio was again used to track the state transition to the “waitingForHCP” state. Checks, similar to those discussed in this section, are again made, along with an error-catching predicate implemented on the waitingForHCP state. The error-catching predicate verifies the task status is one of the permitted outcomes, if not, an error message email is sent to the assignee and the workflow remains in the same state. The predicate code is similar to that shown in Figure 5.4.

5.7. Conclusions

Data integration is realised through implementation of design using MOSS BDC. Profile data is made from local data combined with data from a disparate source through descriptive mark-up in XML. Information targeting is implemented through security, filtering and web parts, although deviation from the original design is required to accomplish functionality. Workflow is implemented by manipulating MOSS lists through Visual Studio 2008 Integrated Development Environment and .Net 3.0 class libraries using the C# programming language.

6. Evaluation

6.1. Introduction

This chapter examines the implementation in terms of its ability to the meet the objectives of: reducing complexity through data integration; increase visibility by providing relevant, secure information to users based on role; and reduce administrative overheads through workflow. In additional this evaluation identifies tradeoffs between the intended design and actual implementations. The system is evaluated in terms of capacity outlining some of the published boundaries of MOSS. It also examines the implemented system in terms of its patient-centric functionality with critical analysis from HCP and IT researchers, obtained from recorded interviews. Full transcripts of interviews are available in Appendix I.

6.2. Data Integration

The aim of data integration in this context is to take data from a disparate source such as an EPR system and use it for more than its designed purpose. A patient record is made up using attributes from disparate sources and combining them with system attributes such as an email address. Data integration is successfully achieved through MOSS's BDC using XML. In terms of the aspect of actually consuming disparate data, there are no design tradeoffs, as all the data intended for consumption was consumed and the MOSS framework behaved as expected from both review and design. Prof. Buchanan comments on the implementation of disparate data integration:

> *"The biggest problem found in healthcare is trying to link to different disparate databases for certain types of research work. This seems to be a way that integrates into a common framework...." (Buchanan, 2008: See Appendix I.2)*

Integrating data into the user profile was successful; however, disadvantages were realised as restrictions in the manipulation of profile information in the context of who could see patient attributes. This meant that a custom security model could not be applied at this level. This limited the use of the returned data as the patient attributes could not assigned to HCPs in a role-based context. This means anyone accessing a patient's My Site could potentially see the whole record. One way to overcome this would use the BDC data in a WSS 3.0 list or a BDC web part. In this case, security models can then be applied. One advantage of this method, however, is that the profile attributes of a patient can be controlled by the patient, allowing the patient to choose which attributes can be seen. The concept of giving patients control of their health records in this way is being considered by Canada and the Netherlands (Rothstein, 2008).

6.3. Information Targeting, Security and Relevance

When implementing information targeting to increase visibility and show information in a role-based context the outcomes are successful in terms of the evaluation of end results. This

is reflected in Dr. Thuemmler comments when asked about the visibility of the system:

> "*...visibility is better. It enables the patient by the end of the day to add and contribute to his own management, to improve the management of his problems which I think makes work easier for all parties involved." (Thuemmler, 2008: See Appendix I.1)*

The implementation, however, did not reflect the perceived design. Minor design trade-offs were identified and implemented to achieve the same functionality. Information from lists stored in a logical department-based site structure was consumed via a single interface expressed as a WSS 3.0 web part page. Web parts containing lists such as appointments were intended to be added to the screen and filtered via connecting web parts based on the users identity built on top if a static access security matrix which identified role. When a list is created in WSS, a web part is automatically created showing a view of the list. It was discovered, during the implementation, that this was true, but the use of the web part was limited to the site in which it was created. This meant that to achieve the required functionality; one of two design changes would be required. The first trade-off would require that the list architecture be changed to a flattened model whereby all the departmental lists reside inside the TLS. This, in effect, destroys the location-based information model. This was implemented, and required a small amount of effort. The implementation is described in Appendix F.1, however, the compromise seems too expensive at the cost of the site structure design and the general information architecture.

The second option was to use a different kind of web part and express list information through direct list query, and manipulating the returned list XML through eXtensible Stylesheet Language (XSL). This option is more faithful to the base design, preserving the information architecture, but had a negative on time scales through extra development. In addition, the suitability of the solution implemented in this way is in doubt as it was discovered through implementation, that these kinds of web parts have a detrimental effect on performance when filtered using personalised methods. This is because the filter cannot cache possible outcomes. It would interesting to establish the exact effects on performance to incorporate into feasibility in terms of scalability.

6.4. Workflow

The workflow implementation, in principle, achieves its goal of reducing administrative overheads, and the implementation excludes high-risk systems such as the postal network and does not require any paperwork to process the patient or log a patient's progress. In terms of a production environment use, the workflow could not be implemented in its present state. Developing a state-based workflow represented the most significant challenge in the project. Workflow technology is relatively new, and most developers opt for a sequential workflow as it represents most business processes. This is echoed in the available literature on the technology (Mann, 2007) (Shelton, 2007). The principle design of workflow was intended to be abstract enough to apply to any referral process, however, this was not achieved as a lack of knowledge of the intricacies of .NET 3.0 workflow object model which prevented complete abstraction. In real terms, aside from the lack of real data types and processes discussed in Chapter 1, the workflow would be applicable to the surgery department. The application of

the dynamic security model as discussed was also not achieved as there was a failure to implement an object that governed the task item permission allowing only the person to whom a task is assigned, access to the item. Rectifying these issues is certainly possible (Alirezaei, 2007), but in the context of demonstrating an approach, the level of development is sufficient. Dr. Thuemmler comments, after the demonstration of this approach.

> *"I think the system will be very helpful, not only for nurses but I think for doctors as well and we can avoid duplicity in all these kinds of things, so I think it is a good thing." (Thuemmler, 2008: See Appendix I.1)*

The unsuitability for production use is also reflected by the lack of interface design and implementation, and the effect of this is that complexity is increased as opposed to reduced. Further development is required to bring this workflow into a scalable domain. The use of forms technology would simplify the interfaces used by only including the functionality required to make the workflow progress. For example the stages where the HCP and patient are required to respond to an appointment request could include a form that only contained the appoint time, static details, comment field and buttons which a trigger accept or reject actions (Janus, 2007). In additional to this, known issues in the MOSS system at the time of development prevented the delay activity, discussed in Section 5.6.3, from being implemented, this effectively meant that timing constraints were implied. Since the start of this project these bugs have been fixed and it is now possible to introduce the delay activity over the "Due date" attribute. An e-mail activity would have been used to remind users that they has not completed their given tasks by the due date set in the program code Appendix **Error! Reference source not found.**.

6.5. Wider Issues

When evaluating the overall design in terms of the strategies discussed in Chapter 2, a number of aims are met in addition to the objectives discussed in Chapter 1. Generally the system is designed to allow access to patient records through the data integration model and display through a user web site. The workflow mechanism allows for an electronic out-patient patient booking and electronic referral recording, at least in terms of event recording. Workflow mechanisms implemented do not explicitly send test results as defined in (Department of Health, 2000) via email (See Section 2.2.2), but communication is established in the workflow implementation via email, therefore it is reasonable to assume that the same approach could be applied to the business processes surrounding patients and test results. All of these features of the prototype address strategies outlined in the NHS National Plan and then NPfIT. Further logging of each step of a workflow can also allow auditor type HCP roles to derive management information through daily tasks, this one of the goals of the IM&T 1992 strategy. In terms of connection for GPs, the system can run on, or off, NHSnet, and communication can be secured using a Secure Sockets Layer (SSL) encryption certificate with little extra connection dependencies except the allocation of sites, groups and user accounts.

6.6. MOSS Boundaries

Using information from (Curry *et al.*, 2008) it is possible to examine this system in terms scalability. This is important information for taking these implementations forward and looking viability in realistic terms. Acceptable performance is defined by Microsoft as 3 seconds or less round trip for common operations such as browsing (Curry *et al.*, 2008); at these levels the software boundaries for the implemented system allows for 250,000 department sites, including the sites created in this prototype; and a limit of 2,000 lists per department site. Each of the implemented lists can hold five million items if accessed programmatically, as demonstrated in the workflow implementation. In terms of users, the limits imposed show that two million users can be in a group such as those defined in Section 5.6.2 and give a total of five million users per profile database. In terms of security and permissions on the system, 2,000 security principles can be applied per department site. These figures represent tested extremes and they are however bound by many other factors, such as the time it would take to restore 250,000 sites in a disaster scenario. In addition, the security principles rely heavily on the directory infrastructure and its ability to cope with millions of security queries. Perhaps the most critical factor in the examination of these figures is that the hardware boundaries of MOSS cannot support them in a practical sense. For example, the hardware configuration for this system could only support approximately 1,000 list items before a loss in performance. The Microsoft System Centre Capacity Planner (SCCP) was used to assess hardware requirements for a national farm, however, it was discovered that Microsoft limits the planner's user capacity to 100,000. If this is a reality, the idea of using My Sites would require approximately 680 farms UK-wide, 11,500 front-end servers and 5,500 databases, each of these elements also have their own limitations. Hardware boundaries are also governed by the type of activity on the system. If a patient is accessing a web page to view it, this has less of an effect on the system than if a patient is updating a task.

Concurrency and usage has the most effect on the system, concurrency is the expected maximum percentage of total amount of MOSS users logged into the system at any one time (English, 2007) and usage is the expect amount operations on the system per second. High concurrency, coupled with high usage, has the most negative effect on performance (Curry et al, 2008). Considering all of these factors makes it difficult to assess capacity of MOSS in very large user context, and to consider a system in terms of hardware requirements requires research into HCP operational activities and expected concurrency figures.

6.7. Patient-Centric

Patient-centric design involves the gathering of information from existing IT systems and the inclusion of patients in the use of a system (Department of Health, 2000). The system design involves the patient in every aspect, and visibility implementations are designed to help provide an instant view of a patient's status and the people that are caring for them.

> *"I think it is very important that patients get a relatively clear view on what's happening. At the moment, it is nearly impossible for patients to oversee in real time what is actually going on with their care and where they stand." (Thuemmler, 2008: See Appendix I.1)*

Workflow designs in this system rely on the inclusion of patients and data integration design is based on a patient record. In both interviews the subjects were asked if they think the system is patient-centric and why:

> *"The patient is definitely strategically placed closer to the centre than we see in the current systems. I think for the time being we can be happy with the position of the patient in the system." (Thuemmler, 2008: See Appendix I.1).*

> *"I think it is definitely patient-centric. It's centred around whoever logs in and they see the data that's relevant to them, so even the healthcare practitioner actually feels that data is revolving around them and should make things easier."(Buchanan, 2008: See Appendix I.2)*

Given this evaluation and the comparison of existing definitions, the system can be considered as patient-centric. My Site is useful in terms of providing an interface for patients to use and interact with the system. One design improvement could be to include all of the data seen on the patientlookup.aspx page, inside the My Site. However, it was discovered early-on in the analysis that My Site represents a disparate site collection with no way of showing the data from the various lists in this way. One way to overcome this would be to adopt the approach used in the implementation of workflow whereby data is copied into the patient site whenever a workflow completes; thus building a patient's site list to accomplish this desired functionality.

6.8. Other Observations

It is worth noting that the implementations of data integration and targeting information to increase visibility are accomplished without the use of program code. This provides leverage over the data layer of the system effectively allowing users to manipulate the data without the training required to interrogate a database directly. In real terms, a HCP or patient can construct their own information environment without waiting or relying on a database programmer to accomplish their information management goals.

7. Conclusions

7.1. Critical analysis

Alleviating the stress of bureaucracy on HCPs allows more time for patient care in both the primary and secondary care bodies within the UK. Making patients more central to healthcare processes is one of the aims of the latest national healthcare strategies. Gathering data from many sources combined with integration and secure presentation, along with explicit inclusion and participation of patients, are characteristics of a patient-centric system (IBM, 2006) (Department of Health, 2000). It has been shown through review that some systems in use today do not have some of these characteristics, whereas others cannot, due to design requirements in place around the time they were created.

This project shows approaches to data integration, role-based, personalised information targeting and security, and workflow using the MOSS information management framework. The system is realised through analysis, design and implementation. Each approach has been proven through implementation and peer review that they could provide some usefulness in further research for application in healthcare environments. Implemented approaches address some of the key issues faced by NHS primary and secondary care bodies and can offer potential solutions such as patient access to healthcare records and electronic out-patient booking. The implemented system has shown that data from a disparate source can be consumed. The usefulness of the consumed data with less than expected in the applied design, but it is noted that outside of this design the data gathered in this way can be used in many other ways (Silversands Consultancy, 2007) (Duerden, 2007).

Relevance was increased through personalised views allowing patients and HCPs to potentially view pertinent information from many sources through a single interface. Implementation of personalised views revealed design problems that were solved through the use of alternative functionality. Workflow implementation shows that it is possible to realise a business process electronically, assign tasks to both HCP and patient, log activities automatically and potentially manipulate security around a patient. The overall system has been considered as patient-centric through peer review, however, more research into these approaches is required as no actual healthcare data or hierarchical security constructs are implemented. Implementation shows that some aspects of the designs may not be appropriate in terms of scalability and performance. This is characterised by the use of non cacheable filter elements for personalised views of list data. Elements of the implementation address some NPfIT requirements such as online booking and access to patient records. Other NPfIT requirements such as blood test results by email, can be achieved through the application of the same approaches in a different context.

Analysis, implementation and evaluation shows that the MOSS framework is tangible in healthcare environments. This is underpinned by the inclusion of MOSS for consideration in other works (Duerden, 2007) in the NHS, and other global healthcare organisations (Kopt, 2006). Capacity planning is difficult to accomplish, available tools are inadequate to accurately plan for more than 100,000 users, yet published software boundaries indicate that MOSS is capable of sustaining 50 times this amount. Therefore, the only way to accurately plan for more than 100,000 users, is through experimentation. Some constraints of MOSS,

such as cross site security, prevent implementation of design facets such as the direct expression of lists in other sites. A workaround is possible through alternate information pathways, however, this could be at the cost of performance. The approach of data integration is achieved by leveraging the MOSS BDC and indexing features. Through this implementation it is reasonable to assume that other data types and sources could be used in the same way. This could extend the reach of HCPs by making information from extended sources such as social work and child services. This could help in diagnostic decision-making, but consent could be offered to the patient as to whether this was allowed or not.

Increasing visibility is achieved through MOSS profiles, security, and WSS 3.0. Using personalised views, this implementation shows that it is possible to have information that is both patient-centred and HCP-centred, residing in the same place or page. The reduction of administrative overheads can be achieved through creating custom MOSS workflows using .NET 3.0 and MOSS programming object models. Implementation shows that paperwork could be reduced by automating manual logging practices. This could help patients by allowing HCPs to spend more time on care within practices. Using workflow to manipulate security models can aid in matters of information security by changing a permission structure around a patient. This can facilitate a need to know medical principles. Mobile access to workflow tasks can make bedside service or mobile medical care less complex, less error prone and timelier thus increasing efficiency.

In conclusion, this system has shown to be of use in healthcare research. The outcomes have provided insight into the possibility of further uses for the approaches used and the implementation of the MOSS framework has shown that rapid application development is possible for a very large-scale system. Some current systems reflect a symptomatic approach to IT. This system, in the gathering of disparate information, workflow and role based targeting can considered as reflecting a holistic approach to patient care.
Can you pick-out some of your main findings from your evaluation ….
What about improving the flow of information from primary and secondary care?

7.2. Future work

Methods used to determine hardware requirements for large-scale implementations require information on concurrency and usage, therefore more research is required into NHS business processes in terms of time, motion and human IT interaction. MOSS software boundaries (Curry *et al.*, 2008), imply that the storage of 68 million summary records similar NHS CRS spine is easily accomplishable, research into using MOSS for such a task may be of benefit for those healthcare bodies that wish to implement such systems. A review shows that MOSS can support HL7 message types through XML forms technology. It may of benefit to research MOSS content types in respect to the HL7 v3 RIM to establish if direct mapping is possible as this would allow the integration of HL7 messages which, in turn, could benefit from other MOSS features such as workflow, indexing, targeting and storage.

Hardware boundaries make it difficult to host personal space for every member of the UK, however, future processing speeds memory capacity, storage sizes and advances in distributed computing may make this possible. Just as interaction between patients and healthcare organisations through MOSS My Sites has been shown to be possible here, the concept of the personal space could be extended as far citizens interacting all aspects of government. This

could include parking tickets, tax, court matters, healthcare and driving, and so on. Right now, a system like this may be of little use as too many people do not use computers; however, a computer literate generation could be interacting like this in years to come.

8. Bibliography

References

Books

Allen, R. & Lowe-Norris, A. (2003) Active Directory, Sebastopol, O'Reilly.
Curry, B., English, B. & Micorsoft Development Teams (2008) Microsoft Office SharePoint Server 2007 Best Practices, Washington, Microsoft Press.

English, B. (2004) Microsoft Sharepoint Products and Technologies Resource Kit, Washington, Microsoft Press.

English, B. (2007) Microsoft Office SharePoint Server 2007 Administrator's Companion, Washington, Microsoft Press.

Hillier, S. (2005) Advanced SharePoint Services Solutions, Berkley, Apress.
Hoffman, K. & Foster, R. (2007) Microsoft SharePoint 2007 Development Unleashed, Indianapolis, SAMS.

Holiday, J., Alexander, J., Julian, J., Robillard, E., Schwartz, B., Ranlett, M., Attis, J. D., Buenz, A. & Rizzo, T. (2007) Proffessional SharePoint 2007 Development, Indianapolis, Wiley.

Janus, P. (2007) Pro Infopath 2007, Berkeley, Apress.
Katz, R. H. (1993) Contemporary Logic Design Reading, Addison Wesley Publishing Company.

Mann, D. (2007) Workflow in the 2007 Microsoft Office System, Berkeley, Apress.
Pattison, T. & Larson, D. (2007) Inside Microsoft Windows Sharepoint Services 3.0, Washington, Microsoft Press.

Walther, S. (2006) ASP.NET 2.0 Unleashed, Indianapolis, SAMS.
Wodtke, C. (2003) Information Architecture Blue Prints for the Web, Indianapolis, New Riders.

Duerden, P., (2007), Our reasons for considering SharePoint 2007 for our information management requirements. **In:** Meeting the Challenges of Information Management in the NHS. London

Freudenstein, P., Gaedke, M., Majer, F. & Nussbaumer, M., (2007), A Workflow-Driven Approach for the Efficient Integration of Web Services in Portals. **In:** IEEE International Conference on Services Computing (SCC 2007). scc, scc.

Parisot, C. & Solomon, H., (2005), DICOM Exchanging Imaging Data. **In:** DICOM Conference 2005. Budapest.

Singureanu, I. (2005). The Evolution of HL7 From V2 to V3. Windham, HL7 Cananda.

Government Documents

Department of Health, (2000). The NHS Plan. Crown Copyright [Cm 4818-I]

Depatment of Health, (1998). Information for Health. Department of Health PO Box 777 London SE17 6XH: Crown Copyright [A1103]

English Department of Health, (1992). Getting Better with Information. An Information Management & Technology Strategy for the NHS in England. London: Crown Copyright

Scottish Consumer Council (2007) How to see your Health Records, Glasgow, Scottish Executive.

Journal Articles

Amy, D., Rosemary, K., Robert, H., Ida, M. A., Patricia, B., Carol, M. C., Susan, J. G., Marcelline, R. H., Susan, M. & Daniel, R. (2003) Modeling nursing interventions in the act class of HL7 RIM Version 3. Journal of Biomedical Informatics, **36,** pp. 294-303.

Fairway, M. (2000) The NHS Plan and Information Technology. BJHC&IM-British Journal of Healthcare Computing & Information Management, **17,** pp. Editorial.

Fougère, M. & Mérette, M. (1999) Population ageing and economic growth in seven OECD countries. Economic Modelling, **16,** pp. 411-27.

Hopper, A., Peifer, A. & Sudduth, B. (1998) A Patient-Centric Approach to Telemedicine Database Development. STUDIES IN HEALTH TECHNOLOGY AND INFORMATICS, **50,** pp. 67-72.

Indrajit, I. & Verma, B. (2007). DICOM, HL7 and IHE: A basic primer on Healthcare Standards for Radiologists. *Indian Journal of Radiology and Imaging.* 2

Karlsson, J. & Eklund, P., (2001) Workflow Design as a Basis for Component Interaction Book MEDINFO 2001 Publisher IOS Press.Volume Volume 84/2001 pp. Pages 1158-60

Kush, R., Alschuler, L., Ruggeri, R., Cassells, S., Gupta, N., Bain, L., Claise, K., Shah, M. & Nahm, M. (2007) Implementing Single Source: The STARBRITE Proof-of-Concept Study. Journal of the American Medical Informatics Association, **14,** pp. 662-73.

Malamateniou, F., Poulymenopoulou, M. & Vassilacopoulos, G. (2003) Emergency healthcare process automation using workflow technology and web services Medical Informatics and the Internet in Medicine, **28,** pp. 195-207.

Rothstein, M. A. (2008) Canada and the Netherland may give patients complete control over health records. Scientific American, **Special Edition,** pp. 69.

Russello, G., Changyu, D. & Dulay, N. (2008) Consent-Based Workflows for Healthcare Management. Policies for Distributed Systems and Networks, **2,** pp. 153 - 61.

Personal Communication

Pusey (2008). History of Oasis. IN Symes, M. (Ed.) London, Oasis Medical Solutions.

Thuemmler, C. (2008). Scenario for Helthcare Process. IN Symes, M. (Ed.).

Thuemmler, C. (2007). Database extract. IN Symes, M. (Ed.).

Web Resources

Balsam Healthcare Corporation (2006). OASIS Implementation Methodology. Available from: http://www.balsam-tech.com/Solutions/implementation.htm [accessed 13, 03, 2008]

BBC (2006). Fears over patient record system. Available from: http://news.bbc.co.uk/1/hi/health/4802560.stm [accessed 26, 06, 2008]

Benson, T. (2007). HL7 v3. Available from: http://www.hl7.org.au/HL7-V3-Resrcs.htm [accessed 15, 04, 2008]

Bowers, S. (2006). Accenture to quit NHS technology overhaul. Available from: http://www.guardian.co.uk/technology/2006/sep/28/news.business [accessed 08, 06, 2008]

Brindley, M. (2007). GPs quit NHS over paperwork. Available from: http://www.walesonline.co.uk/news/health-news/2007/10/22/gps-quit-nhs-over-paperwork-91466-19987395/ [accessed 03,05, 2008]

Clark, L. (2007). Milton Keynes hospitals go live with electronic care records. Available from: http://www.computerweekly.com/Articles/2007/02/28/222112/milton-keynes-hospitals-go-live-with-electronic-care.htm [accessed 25, 08, 2008]

Colins, T. (2008). Patients left untreated at Barnet and Chase Farm Hospitals NHS Trust. Available from: http://www.computerweekly.com/Articles/2008/08/13/231806/patients-left-untreated-at-barnet-and-chase-farm-hospitals-nhs.htm [accessed 20, 08, 2008]

Cooper, J., Urquhart, C., Wailoo, A. & Yeomen, A. (2001). NeLH Pilot Evaluation Project. Available from: http://www.dil.aber.ac.uk/dis/research/reports/nelh-pilot-evaluation-report6.pdf [accessed 20, 07, 2008]

Cordonnier, E. (2002). DICOM and the Web. Available from: http://medical.nema.org/dicom/koreanpacs/DICOM_and_the_WEB_by_E_Cordonnier.ppt [accessed 23,05, 2008]

Davies, J. (2008). NHS Trust transforms security and access management. Available from: http://www.computeractive.co.uk/computing/news/2219321/nhs-trust-transforms-security [accessed 20, 06, 2008]

Dobson, R. (2000). Net losses. Available from: http://www.communitycare.co.uk/Articles/2000/05/26/8685/net-losses.html [accessed 12, 04, 2008]

Doward, J. (2008). Chaos as £13bn NHS computer system falters. Available from: http://www.guardian.co.uk/society/2008/aug/10/nhs.computersystem [accessed 10,08, 2008]

E Health Media (2005). North-west GPs to view results over NHSnet. Available from: http://www.e-health-insider.com/news/1098/north-west_gps_to_view_test_results_over_nhsnet [accessed 12, 04, 2008]

Evans, J. (2008). Library & Information Service Newsletter. Available from: http://www.nbt.nhs.uk/researcheducation/library/Newsletters/May%2008.doc [accessed 21, 06, 2008]

Foot, T. (2008). Hospital computer 'chaos' Available from: http://www.thecnj.co.uk/camden/2008/070308/news070308_09.html [accessed 26, 07, 2008]

Fulop, N., Hendy, J., Hutchings, A. & Reeves, B. (2005). Evaluation of the implementation of the NHS Care Record Service (NCRS). Available from: http://www.sdo.nihr.ac.uk/files/adhoc/conference-2005-hendy.pdf [accessed 22,05, 2008]

Greening, C. (2004). Provision of ICT Services to Worcestershire General Practices. Available from: http://www.worcestershirehealth.nhs.uk/EXTRANET_Library/pcimt/gms/files/04_07_26/gms_n_040726_03.doc [accessed 27,05, 2008]

IBM (2006). Patient-centric: the 21st century prescription for healthcare. Available from: http://www-03.ibm.com/industries/healthcare/doc/content/bin/Patient_Centric_wp.pdf [accessed 06,07, 2008]

Indigo 4 (2007). Electronic Clinical Requesting and Reporting in Primary and Secondary Care. Available from: http://www.indigo4.com/brochures/RequestingReportingPriSec.pdf [accessed 17, 04, 2008]

Indigo 4 (2004). Keystone Classic and Keystone Enterprise. Available from: http://www.indigo4.com/keystone.php [accessed 25, 04, 2008]

LeadTechnologies (2007). Programming Interfaces. Available from: http://leadtools.com/SDK/Medical/Medical-Imaging-Suite.htm [accessed 21, 05, 2008]

Leeds & Bradford Pathology Partnership (2007). Full Blood Count Available from: http://www.pathology.leedsth.nhs.uk/pathology/ClinicalInfo/CommonTestsInvestigations/FullBloodCount/tabid/130/Default.aspx [accessed 21, 03, 2008]

Mackenzie, B. (2004). A history of NHS IT to the Millennium. Available from: http://www.bioneural.net/2004/04/29/a-history-of-nhs-it-to-the-millennium/ [accessed 09,04, 2008]

MillarHutchinson (2000). NHSnet Addressing User Perceptions. Available from: http://www.infocus-nw.nhs.uk/file/pdfs/NHSnet%20-%20Addressing%20user%20perceptions.pdf [accessed 22, 05, 2008]

Montalbano, E. (2008). Microsoft SharePoint powers Olympics Portal. Available from: http://www.pcworld.com/businesscenter/article/149541/microsoft_sharepoint_powers_olympics_portal.html [accessed 20,08, 2008]

Nation Electrical Manufacturers Association (NEMA) (2007). Digital Imaging and Communications in Medicine (DICOM). Available from: ftp://medical.nema.org/medical/dicom/2008 [accessed 21,05, 2008]
Royal College of Nursing (2008). RCN survey shows nurses spend more than a million hours a week on paperwork. Available from: http://www.rcn.org.uk/newsevents/news/article/uk/rcn_survey_shows_nurses_spend_more_than_a_million_hours_a_week_on_paperwork [accessed 01, 05, 2008]

Shelton, R. (2007). Building a Multilevel Approval Workflow with SharePoint (MOSS 2007) and Visual Studio. Available from: [accessed 23, 05, 2008]

Silversands Consultancy (2007). http://www.silversands.co.uk/images/uploads/Napier%20University.pdf. Available from: http://www.silversands.co.uk/images/uploads/Napier%20University.pdf [accessed 12,10, 2007]

Telemedicine (2008). AeroScout Announces Agreement with Hill-Rom to Deliver Healthcare Visibility Solutions to Hospitals Nationwide. Available from: http://www.3dtelemedicine.com.au/casestudies/aeroscout_feb2608.asp [accessed 24, 02, 2009]

Wray, R. (2006). iSoft in crisis over £6bn NHS project. Available from: http://www.guardian.co.uk/society/2006/jul/07/epublic.politics [accessed 28,07, 2008]
Young, T. (2008). NHS IT may face additional delays after Fujitsu is ditched. Available from: http://www.computing.co.uk/computing/news/2218287/nhs-face-additional-delays-4041632 [accessed 06,06, 2008]

Bibliography

Davies, J. (2008). NHS Trust transforms security and access management. Available from: http://www.computeractive.co.uk/computing/news/2219321/nhs-trust-transforms-security [accessed 20, 06, 2008]

McCafferty, C. (1996) Securing the NHSnet. BJHC&IM-British Journal of Healthcare Computing & Information Management, **13,** pp. 24-6.

NHSIA (2000). Post-NHSnet Connection pack. Available from: http://www.doctoronline.nhs.uk/masterwebsite1Asp/targetpages/nhsNet/NHSNet.htm [accessed 23,05, 2008]

Office_for_National_Statistics (22, 02. 2005). NHS hospital waiting lists: by patient's region of residence and NHS Regional Office area, at 31 March 2002: Regional Trends 37. Available from: http://www.statistics.gov.uk/StatBase/Expodata/Spreadsheets/D5939.xls [accessed 03, 06, 2008]

Office_for_National_Statistics (09, 03, 2004). NHS Hospital and Community Health Service staff: by type of staff and NHS Regional Office area, 30 September 2002: Regional Trends 38. Available from: http://www.statistics.gov.uk/StatBase/Expodata/Spreadsheets/D7780.xls [accessed 06, 05, 2008]

Scottish National Health Service (2004). National ehealth/IM&T Strategy 20042008 Available from: http://www.show.scot.nhs.uk/imt/WordDocs/National%20eHealth%20IMT%20Strategy%20April%2004%20final%20draft.doc [accessed 23, 03, 2008]

Appendix 1

Initial Project Overview

Title:

Distributed healthcare framework using patient-centric role-based security modelling and workflow.

Overview of Project Content and Milestones

To create a system by which healthcare professionals may consume data from disparate sources via an integrated dashboard interface. To model practical patient processes in healthcare organisations into a digital workflow environments using a rapid application development frameworks.

Milestones

In order to successfully achieve the above summaries; a number of key objectives must be completed including the visualisation of patient movements within a healthcare environment and identification of the key processes used by healthcare professionals in primary, secondary and tertiary care. The establishment of dialogue with healthcare research representatives in order to identify the areas in which access to disparate data sources would have a practical and demonstrable application. The above objectives are required to move forward in the project and would be expected to be completed within 8 weeks of project commencement.

Literature review; it is expected the literary reviews are to be completed to draft levels by week 12 of the project

Equipment procurement, installation and configuration of hardware / software platforms should be completed by week 16

By week 20; a working workflow model should completed as well as integrated access to disparate database systems via dashboard technology.

This should leave 7 week period to allow for the completion of documentation including poster sessions and provide a small amount of headroom for project slippage including the possible comparison of other workflow and data models.

The Main Deliverable(s)

To provide a proof of concept for healthcare researchers commissioned to investigate UK and EEC directives outlining requirements for research into patient-centric data modelling and role-based security as part of an advanced healthcare framework.

To demonstrate workflow modelling technology which extend patient-centric concepts by providing possible solutions to wider issues faced by UK and European healthcare organisations.

To provide a working, tangible distributed system prototype incorporating the stated deliverables for demonstration to healthcare researchers using technology that can build onto existing healthcare infrastructures.

The Target Audience for the Deliverable(s)

The main audience for deliverables in this project are healthcare research representatives, interested individuals conducting similar investigations, doctors, nurses and hospital administrative agents.

The Work to be Undertaken

Research government and EEC directives into patient-centric data modelling
Liaise with healthcare researchers to discuss the current requirements
reduce the scope of the main problem in order to contain the project within manageable boundaries.
Investigate geo-dispersive capabilities of new distributed database technologies
Analysis of business processes pertaining to patients and the human roles involved in processing said patients.
Define roles based on previous analysis.
Define a security model based on the previously defined roles.
Define a number of test scenarios based on information obtained from liaisons with healthcare researchers.
Create a set of test users based on defined roles.
Procure resources for implementation of distributed infrastructure.
Build distributed infrastructure including data stores, core directory services and user interface servers.
Implement defined security model within authentication and authorisation infrastructure using test users.
Define a single basic workflow process mapping a real life business process within the organisation based around the patients' activity and information pathways.
Develop custom electronic workflow using basic workflow definition
Design, build and implement patient-centric framework in a software environment.
Implement electronic workflow
Test and evaluate using predefined scenarios
Demonstrate framework to healthcare researchers

Additional Information / Knowledge Required

Extending current skills in information management technologies using features in the "Microsoft .Net" version 3 development framework and in particular; learning and developing custom workflows using Microsoft Visual Studio 2005. Further implementation of new technologies which provide the ability consume data from disparate data sources by using Line of Business (LOB) instance elements in XML.

Knowledge is required to try and understand role based data access activities and investigation into the understanding of patient based information pathways in healthcare practices is required to achieve successful patient based workflow modelling.

Techniques in custom workflow programming using .NET 3 framework are required to technically achieve the successful implementation of a workflow model.

Information Sources that Provide a Context for the Project

This section contains examples of the kinds of information sources that will be used for the project. It is by no means a definitive bibliography.
Internet sources that will be used include: http://www.acm.org, http://www.sharepointu.com,

Books include

Mann, D. 2007. *Workflow in the 2007 Microsoft Office System.* USA: Apress

Conferences proceedings include

Bennett, K. Brereton, P. Budgen, D. Keane, J. Kotsiopoulos, I. Layzell, P. Rigby, M. Russell, M. Turner, M. Zhu, F. 2004 Using Web Service Technologies to create an Information Broker: An Experience Report. *Proceedings of the 26th International Conference on Software Engineering.* The Computer Society

Ferrin, D. Miller, M. Giron, G. 2000 Electronic Workflow for Transaction-Based Work Cells in a Financial Services Firm. In J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick (Eds.) *Proceedings of the 2000 Winter Simulation Conference:* Ferrin, Miller, and Giron pp. 2055-2058

The Importance of the Project

This project attempts to address some of the key challenges faced by UK and European healthcare organisation. The outcomes from this project will provide much needed data for continuing research into IT solutions for applications in healthcare environments.

The Key Challenge(s) to be Overcome

The overall success of this project can be measured equally by the success or failure of the data model used as in proof of concept; failures are equally as important as successes. Outcomes and evidence obtained during this project will provide a platform or basis for continuing research. One Key point of failure would be the unsuccessful implementation of any data model preventing any comparative analysis from taking place.

Appendix 2 Project Diaries

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 26/02/2007 **Last diary date: N/A**

Objectives:

- Provide supervisor with work history, programming background, a history of my project involvements, my academic achievement records and case study participation
- Discuss with supervisor possible involvement with a number of projects based on law enforcement, healthcare initiatives, information management and existing investigations
- Discuss with supervisor any research papers that should be obtained for reading literature review

Progress:

- Relayed all information that there was time for and agreed to supply more information by email to Professor Buchanan.
- Agreed to wait for next supervisor meeting to finalise project title in order to give Professor Buchanan time to digest the supplied information and confer with colleagues on existing research initiatives
- Agreed to start looking on ACM portal at subject titles of 'healthcare' and 'data modelling'

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 12/03/2007 **Last diary date: 26/02/2007**

Objectives:

- Review existing research materials provided by supervisor based on information supplied in the previous meeting
- Decide on project title
- Begin project overview document

Progress:

- After reviewing materials of existing research initiatives provided by Professor Buchanan further discussions, it was decided that the issue of healthcare in relation to information management and digital workflow where the fields of study where previous actitities would be most beneficially applied
- Agreed with Professor Buchanan that the scope of the project was large and discussed ways to narrow the field of study to that which is indicative of an honours project.

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 19/03/2007 **Last diary date: 12/03/2007**

Objectives:

- Hand in draft project overview document for supervisor review
- Meet with NHS research representatives
- Begin correlation of literature review
- Begin planning for installation of demonstration framework for supervisor and healthcare research representative

Progress:

- Professor Buchanan has taken project overview for review
- NHS research reps were unavailable this week
- Poor progress with literature review
- Gained permission to temporarily use resources for demonstration of new technologies and possible applications to supervisor and healthcare research representative

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 26/03/2007 **Last diary date: 19/03/2007**

Objectives:

- Hand in project overview
- Attend second marker interview
- Obtain literature review advice from supervisor
- Build infrastructure for demonstration

Progress:

- Project overview not complete, waiting for data
- Second marker interview delayed a week due to personal illness
- Obtained literature help from supervisor and previous honours syudents
- Discussed weaknesses in review techniques and search issues
- Completed installation of framework for demonstration purposes

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 02/04/2007 **Last diary date: 26/03/2007**

Objectives:

- Seek permission to use extra resources for project from university IT department.
- Attend second marker interview
- Complete demonstration framework for second marker interview
- Seek IT resources for project submission
- Obtain software required for project.

Progress:

- Permission has been granted for the use of C & IT development domain including an instance of Microsoft exchange server and Active Directory components.
- Second marker interview delayed Until Wed 25 April 2007
- Completed installation of framework for demonstration purposes
- A server has been provided by Prof. Buchanan for the installation of Microsoft Office Sharepoint Server 2007 (MOSS), the main framework in which the project is to be developed.
- An instance of MOSS has been made available by Dr. Jose Munoz via Napier Universities Academic account with Microsoft.

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 04/01/2008 **Last diary date: 07/05/2008**

Objectives:

- Re initiate project
- Procure hardware for self development

Progress:

- Personal circumstances meant that the project was on hold from 04/2007 to 01/2008, in that time a number of changes to Napier University meant that much of the development work had to be restarted although the aims remain the same.
- Supervisor was unavailable this week due to holidays
- Instabilities in availability of provided infrastructure presented a significant risk to the project, therefore decision was made to procure own hardware for project requirements
- Ordered hardware components sufficient to support environment

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 11/01/2008 **Last diary date: 04/01/2008**

Objectives:

- Assemble procured hardware and install operating system
- Download required software from Microsoft Academic Alliance
- Purchase virtual machine software in order to support portability in the project and ease transition in any research handover

Progress:

- Ordered components were delivered and assembled
- Windows 64bit operating system downloaded and installed
- Virtual machine software delivered in installed

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 18/01/2008 **Last diary date: 11/01/2008**

Objectives:

- Build 3 windows 2003 servers in a virtual environment.
- Configure virtual network adapters and router
- Install all required updates and service packs
- Install .net 2.0 .net 3.0

Progress:

- Three servers built
- Virtual adapters configured
- Updates installed
- .net frameworks not installed; configuring network took longer than expected

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 25/01/2008 **Last diary date: 18/01/2008**

Objectives:

- Install .Net 2.0 and .Net 3.0 frameworks on all servers
- Purchase domain name stirlinghealthcare.com
- Configure 1 Microsoft Windows 2003 server to be a domain controller office.stirlinghealthcare.com
- Configure local DNS

Progress:

- .Net installed
- Domain Name purchased and registered
- Domain controller configured successfully
- DNS server required for domain controller was configured, forwarders set up to allow client access to internet

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 01/02/2008 **Last diary date: 25/01/2008**

Objectives:

- Extend Active Directory schema for Microsoft Exchange Server 2003
- Install .net web application server on domain controller
- Install Microsoft Exchange Server 2003 on domain controller
- Test email accounts and web access components
- Arrange meeting with supervisor

Progress:

- .All applications successfully installed

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 08/02/2008 **Last diary date: 01/02/2008**

Objectives:

- Purchase SSL certificate
- Install SSL certificate for www.stirlinghealthcare.com
- Configure router to operate on port 443
- Test secure email communication
- Email supervisor

Progress:

- Received SSL and installed certificate on IIS for Exchange server
- Configured router and prevented access to email over port 80
- Secured web domain
- Tested email
- Emailed supervisor

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 15/02/2008 **Last diary date: 08/02/2008**

Objectives:

- Think about the possibility of using test data instead as advised during week 9 inteview
- Join additional member servers to domain
- Configure prerequisites for installation of Database
- Configure service account for sharepoint and SQL server
- Install configure and test Microsoft SQL Server 2005 on a member server

Progress:

- Absence of contact with supervisor and has put project at risk in terms of obtaining real data sets, at the week 9 inteview, the second marker suggested the possibility of a setting a threshold by which no further effort to obtain real data would be made and the decision to use made up data have to be made, the project has reached this threshold
- The 3 remaining windows 2003 servers were joined to the office.stirlinghealthcare.com domain and configured as member serversv
- Studied installation documentation for SQL server 2005 and made preparations to install
- Installed MSSQL servewr 2005 on a member server
- Tested database communications over named pipes from another server

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 22/02/2008 **Last diary date: 15/02/2008**

Objectives:

- Install web application servers on 2 member servers
- Install .Net 3.5 framework on 2 member servers
- Install visual studio 2008 on one member server
- Install Microsoft Office SharePoint Server (MOSS) on 2 member servers
- Run farm setup and configure a medium farm in MOSS

Progress:

- The remaining unassigned members servers were configured to run MOSS each server was made an application server and .Net 3.5 framework was installed.
- 1 server was configured for SharePoint code staging platform by installing Visual studio 2008 on the server intended to be a SharePoint web front end
- MOSS was then installed on 2 member servers
- A small/medium farm was configured

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 29/02/2008 **Last diary date: 22/02/2008**

Objectives:

- Assign services to farm servers
- Install service packs on shareppoint
- Define department sites
- Add users
- Discuss product in write up

Progress:

- Having installed the platform infrastructure for sharepoint, the farm was configured SASSPSWFE1 was configured to be the front end
- SASSPSSearch was configured to be search, index and host the Central Administration web site sor sharepoint
- SASSQLDEV01 was the database backend for the product.
- Department web sites were defined but not created as it discovered that services packs containing bug fixes for elements of the sharepoint intended for use in project were required, these service were installed before ant development of the framework took place.
- SahrePoint was discussed in writeup

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 07/03/2008 **Last diary date: 29/02/2008**

Objectives:

- Design site structure
- Design lists
- Establish required site columns
- Review issues present in NHS in terms of HCP, social and technical for context
- Look at existing healthcare IT systems

Progress:

- Looked at issues with NHS systems, doctors nurses and IT opinions of the state of IT in healthcare
- Completed rough design of the required list will probably use a task type list as I will want to implement workflow

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 14/03/2008 **Last diary date: 07/03/2008**

Objectives:

- Implement list design
- Implement Site structure
- Look at other other it systems in healthcare
- Look at communication standards for literature
- Continue to write up and make changes to report

Progress:

- Looked a t HL7 v 2 and v3
- Looked at DICOM
- Looked at Kodak R4
- Met with supervisor and agreed changes to document

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 21/03/2008 **Last diary date: 14/03/2007**

Objectives:

- Investigate HCP roles by looking at Office for national statistics
- Review licensing costs in national terms for possible use in evaluation

Progress:

- Looked at office for national statistics. Will probably use generic terms as representative of HCP roles
- Considering the use of made up data as it avoids ethical issues and doesn't rely on anyone else's input
- Licensing is not tangible as a use in evaluation, VLK exist Microsoft won't release that kind of information, unless I am a government organisation, which I'm not. The best I can manage conjecture at 33% of face price.

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 28/03/2008 **Last diary date: 21/03/2008**

Objectives:

- Write up introduction to methodology
- Add defines user to to Active Directory
- Configure the SSP used for methodology
- Verify Shared Service Provider al ensure My Site functionality#
- Perform a profile import for testing purposes

Progress:

- The basis of the methodologies, such as required apparatus dependencies and site design were written up
- Concerns over the length of the MOSS discussion were noted and a decision was made to move the part not used in direct implementation , to the Appendices
- Noted some other applications of MOSS in the NHS
- Created final SSP and checked the My Site functionality was intact.
- Ensured profile import was working prior for BDC implementation.

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 05/04/2008 **Last diary date: 28/03/2008**

Objectives:

- Learn workflow
- Design BDC tables for data integration
- Design targeting

Progress:

- Read and tried Mann Apress seemed to have achieved sequencial workflow, I require state based apply prior University knowledge and apply appropriate methodologies. Very challenging and time consuming.
- Used an example database table from an EHR system, There were far too many columns for demonstrative purposes so columns representing Blood type, Sex, Ethnic Origin and Current GP were used

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 12/04/2008 **Last diary date: 05/04/2008**

Objectives:

- Tabulate users details in AD for write up
- Create and populate table in disparate source for data integration
- Verify disparate data by connecting directly to it and performing SQL query
- Create a basic Line Of Business instance using XML in an Application Definition File (ADF)
- Design workflow for project
- Upload report containing changes requested by supervisor
- Continue to look at standards for literature review

Progress:

- Wrote up user tables and supplemental information such as discussing the Business Data Catalogue (BDC)
- Created "profile_supp" table in PostGreSQL database server hosted in Manchester
- Populated table
- Created basic LOBInstance in an XML ADF file but did it not work because the Single Sign On (SSO) database in SharePoint was not configured
- Test workflows implemented and tested to ensure moving forward with was possible
- Workflow design using knowledge gained from modules. Basic FSM diagram using a less complex scenario than previously imagined.
- Made changes to report

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 19/04/2008 **Last diary date: 12/04/2008**

Objectives:

- Implement Business Data Catalogue (BDC)

Progress:

- The Single Sign On data base was configured and a successful basic BDC implementation was created demonstrating the connection and consumption of disparate data, however, the methodology requires a BDC implementation as supplement to profile import, having never done this before more investigation into the methods required to accomplish this was required.
- ADF File was loaded and now SSO works, however, methods are missing that allow the Indexer to crawl the disparate source

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 26/04/2008 **Last diary date: 19/04/2008**

Objectives:

- Implement Business Data Catalogue (BDC) –write indexing specific finder methods
- Map data from disparate source to profile attributes
- Write some more

Progress:

- After several iterations of the ADF file a successful import of profile data was recorded, however errors were occurring, extensive examination of trace logs revealed Indexer performing import..
- PostGreSQL drivers were not present on the index server so they are now installed and successful imports are occurring
- Import was successful, however, properties are not being mapped, checked literature, specific finder method required. Implemented SpecificFinder Method. Profile import working as expected.
- MOSS not behaving as expected, cannot apply roles to profile data, patients can only choose what not to allow anyone to see about themselves, gives power to patient but doesn't help in real context

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 03/05/2008 **Last diary date: 26/04/2008**

Objectives:

- Implement targeting designs

Progress:

- Implementing target designs; had issues with perceived list functionality, had to use a different type of web part and some extra XSL or flatten the lists to and under the TLS to achieve the result of targeting the created lists. Went for both options, XSL was more complicated but preserved the structure.
- Wrote additional method

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 10/05/2008 **Last diary date: 03/05/2008**

Objectives:

- Implement workflow designs
- Continue to write up MOSS, Designs and Methods
- Prepare the document for an initial submission for review by supervisor
- Look at NHS strategies in terms of IT

Progress:

- Started layout the FSM in VS2008, then translated design action into MOSS SharePoint activities
- Made changes to report
- Looked at NHS strategies from 1992, 1998, and NPfIT, looked at others, but will probably use these in lit review.

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 17/05/2008 **Last diary date: 10/05/2008**

Objectives:

- Hand in work done so far for review by supervisor
- Arrange meeting with supervisor
- Start to write literature review
- Continue with workflow implementation
- RCN Nurses are complaining of too much paper work; will use in context of report

Progress:

- Successfully implemented the main design, some actions such as email and logging are still to be accomplished but the work does change state and the predicates are working as expected.
- Made changes to report
- Tried to arrange meeting but will just cold call if I can't get hold supervisor
- Started to write literature review

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 24/05/2008 **Last diary date: 10/05/2008**

Objectives:

- Meet supervisor
- Continue write literature review
- Continue workflow implementation

Progress:

- Successfully implemented the main design, some actions such as email and logging are still to be accomplished but the work does change state and the predicates are working as expected.
- Will go to supervisor to try and get feedback.
- Continues with literature review looked at DICOM development software

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 31/05/2008 **Last diary date: 24/05/2008**

Objectives:

- Meet supervisor
- Continue write literature review
- Finish workflow implementation

Progress:

- Continued to add functionality to workflow implementation, tried to implement per item security but more research required, will leave workflow now as most of functionality is achieved, will spend more time on it after more written work is complete.
- Met supervisor and he asked for changes to document layout

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 07/06/2008 **Last diary date: 31/05/2008**

Objectives:

- Make changes to document structure
- Submit changes to document structure
- Continue write literature review

Progress:

- Made significant changes to the report structure, split design and implementation chapters concentrated on context and writing style, review previously gathered material and reworked initial context statement adding references.
- Submitted changes
- Looked at waiting list statistics

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 1/08/2008 **Last diary date: 21/06/2008**

Objectives:

- Continue Documentation
- Make contact with supervisor

Progress:

- Met with supervisor – the project was again disrupted due to both personal circumstances and the availability of the supervisor, himself constrained by the academic calendar and annual leave.
- Required to make more changes to document
- Required to set up interview for evaluation
- Made more changes to document
- Split literature review to form analysis and literature review
- Made appointment with supervisor

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 08/08/2008 **Last diary date: 01/08/2008**

Objectives:

- Continue Documentation
- Meet supervisor

Progress:

- Met with supervisor and agreed to more changes and refinement to the report
- Demonstrated system to supervisor and made contact with Dr. Cristoph Thuelmmer
- Discussed the possibility of peer review through recorded interview
- Began to write critical analysis of system including trade identifying issues encountered

Supervisor's Comments:

NAPIER UNIVERSITY

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 15/08/2008 **Last diary date: 08/08/2008**

Objectives:

- Continue Documentation
- Meet supervisor

Progress:

- Met with supervisor and agreed to more changes and refinement to the report
- Arranged peer review through recorded interview
- Continued to write critical analysis of system including trade offs identifying issues encountered
- Continue reviewing technologies to add weight to report
- Concerned about word count

Supervisor's Comments:

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 15/08/2008 **Last diary date: 08/08/2008**

Objectives:

- Continue Documentation
- Meet supervisor
- Finalise introduction write up
- Finalise design write up
- Finalise figures

Progress:

- Met with supervisor and agreed to include high level design and accompanying diagram
- Peer review through cannot take place until September due to availability of researcher
- Finalised introduction
- Evaluated the system in the context of boundaries
- Finished design
- Finished analysis
- Continued with evaluation
- Still have concerns of word count although the content seems relevant

Supervisor's Comments:

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 22/08/2008 **Last diary date: 15/08/2008**

Objectives:

- Continue Documentation
- Meet supervisor
- Begin conclusion
- Finalise literature review

Progress:

- Met with supervisor discussed word count and conclusions, supervisor agreed to word count and discussed creation of alternative version, it was agreed that is should be arranged.
- Still have concerns of word count although the content seems relevant
- Discovered MOSS indexing used for Olympics; will use in analysis

Supervisor's Comments:

SCHOOL OF COMPUTING

PROJECT DIARY

Student: 03010021 **Supervisor: Professor William Buchanan**

Date: 10/09/2008 **Last diary date: 22/08/2008**

Objectives:

- Conduct interviews
- Meet supervisor
- Finalise document
- Create edit

Progress:

- Interviewed Bill Buchanan and Christoph Thuemmler
- Included the material obtained from interviews in the evaluation chapter
- Discovered Americans are considering implementing similar approach to patient as was attempted in data integration for profile properties, there idea is to give the management of security to the patient, their model also includes roles. Included this information in report.
- Added required Appendices
- Finalised documentation

Supervisor's Comments:

Appendix A

A.1 User tables

These tables refer to Section 5.3.

Table A 1 User table

Username	Name	Role	Department	Location
GP	Mark Thomson	Practical	GP	Medical Practice
Doctor	Cathy Risedale	Practical	A&E	Hospital
HomeCarer	Sue Higgins	Practical	Out Patient	Mobile
Surgeon	Walter Reed	Practical	Surgery	Hospital
SurgeonSecretary	Andrew Miller	Administrative	Surgery	Hospital
GPSecretary	Anne Murray	Administrative	GP	Medical Practice
Researcher	Antony Wilks	Research	Research	Hospital
RadioLab	Gerry Anderton	Laboratory	Radiology	Hospital
CardioLab	Jane Goodhart	Laboratory	Cardiology	Hospital
Patient01	Janet Tailor	Patient	NA	NA
Patient02	Bob Moore	Patient	NA	NA
Patient03	Karen Anderson	Patient	NA	NA

Table A 2 Directory configuration of user accounts

Domain Username	e-mail Address	First Name	Last Name	Display Name	Job Title	Department	Company	Office
OFFICE\GP	GP @stirlinghealthcare.com	Mark	Thomson	Mark Thomson	GP	GP	Grange Medical Practice	Medical Practice
OFFICE\Doctor	Doctor @stirlinghealthcare.com	Cathy	Risedale	Cathy Risedale	Doctor	A&E	NHS	Hospital
OFFICE\HomeCarer	HomeCarer @stirlinghealthcare.com	Sue	Higgins	Sue Higgins	Home Carer	Out Patient	NHS	Mobile

OFFICE\Surgeon	Surgeon @stirlinghealthcare.com	Walter	Reed	Walter Reed	Surgeon	Surgery	NHS	Hospital
OF-FICE\SurgeonSecretary	SurgeonSecretary @stirlinghealthcare.com	Andrew	Miller	Andrew Miller	Surgeons Secretary	Surgery	NHS	Hospital
OFFICE\GPSecretary	GPSecretary @stirlinghealthcare.com	Anne	Murray	Anne Murray	GP Secretary	GP	Grange Medical Practice	Medical Practice
OFFICE\ Researcher	Researcher @stirlinghealthcare.com	Antony	Wilks	Antony Wilks	Researcher	Research	NHS	Hospital
OFFICE\RadioLab	RadioLab @stirlinghealthcare.com	Gerry	Anderton	Gerry Anderton	Radiologist	Radiology	NHS	Hospital
OFFICE\CardioLab	CardioLab @stirlinghealthcare.com	Jane	Goodhart	Jane Goodhart	Cardiologist	Cardiology	NHS	Hospital
OFFICE\Patient01	Patient01 @stirlinghealthcare.com	Janet	Taylor	Janet Taylor				
OFFICE\Patient02	Patient02 @stirlinghealthcare.com	Bob	Moore	Bob Moore				
OFFICE\Patient03	Patient03 @stirlinghealthcare.com	Karen	Anderson	Karen Anderson				

A.2 Application Definition File from Section 5.4.4

XML declaration

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
```

Root node LOBSystem

```
<LobSystem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.microsoft.com/office/2006/03/BusinessDataCatalog BDCMetadata.xsd" Type="Database" Ver-
sion="1.0.0.0" Name="scu_portal" xmlns="http://schemas.microsoft.com/office/2006/03/BusinessDataCatalog">
  <Properties>

   <Property Name="WildcardCharacter" Type="System.String">*</Property>
 </Properties>
```

LOBSystemInstance

This section describes the database connection, authentication methods and any SSO instances used.

```
 <LobSystemInstances>
  <LobSystemInstance Name="scu_portal_Instance">
    <Properties>
     <Property Name="rdbconnection Dsn" Type="System.String">Mammoth ODBCng Beta</Property>
     <Property Name="DatabaseAccessProvider"
Type="Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbAccessProvider">Odbc</Property>
     <Property Name="AuthenticationMode"
Type="Microsoft.Office.Server.ApplicationRegistry.SystemSpecific.Db.DbAuthenticationMode">RdbCredentials</Property>
     <Property Name="SsoApplicationId" Type="System.String">ook</Property>
    </Properties>
  </LobSystemInstance>
 </LobSystemInstances>
```

Entities

This part of the ADF describes the database table and the primary key or keys represented as Identifiers.

```
 <Entities>
  <Entity EstimatedInstanceCount="10000" Name="profilesupp">
   <Identifiers>
    <Identifier TypeName="System.String" Name="pid" />
   </Identifiers>
```

Method

This section describes the operations used to retrieve data from the source, in this example the data source is an SQL database which requires text based commands in the form of SQL statements

```
<Methods>
 <Method Name="Find_profilesupp">
  <Properties>
   <Property Name="RdbCommandType" Type="System.Data.CommandType, System.Data, Version=2.0.0.0, Culture=neutral, PublicK-
eyToken=b77a5c561934e089">Text</Property>
   <Property Name="RdbCommandText" Type="System.String">Select "pid","bldtyp","ethno","sx","cgp" from  profilesupp where pid
like (?)</Property>
  </Properties>
```

Filter Descriptor

To establish a 1:Many relationship with the user attribute "Username" in AD this filter is required, the crawler uses this filter to parse the row in the disparate source during the import process

```
<FilterDescriptors>
  <FilterDescriptor Type="Wildcard" Name="pidFilter" />
 </FilterDescriptors>
<Parameters>
```

Input parameter

Signified it's direction, the input parameter is the variable used to query the disparate database, special syntax is used for each type or brand of database in the case of PostGresSQL "(?)" is used. The syntax allows the disparate database to map a passed variable to column value; in this case, the passed variable is mapped to the pid column.

```
<Parameter Direction="In" Name="(?)">
 <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089"
AssociatedFilter="pidFilter" IdentifierName="pid" Name="pid">
    <DefaultValues>

    <DefaultValue MethodInstanceName="Find_profilesupps_Instance" Type="System.String">%</DefaultValue>
    <DefaultValue MethodInstanceName="Find_profilesupp_Instance" Type="System.String">%</DefaultValue>
    </DefaultValues>

        </TypeDescriptor>
```

Return Parameter

Again; signified by its direction, the return parameter describes retrieval methods and the dataset returned by the disparate source via TypeDescriptors present in child nodes. The first two TypeDescriptors describe ADO.NET data readers, the first of which provides a collection of records and second providing a single row. The remaining TypeDescritpor describe each column of the disparate source. Localised display names can be added to the column TypeDescriptors which display column as more user friendly names.

```
</Parameter>
<Parameter Direction="Return" Name="@profilesupp">
```

```
        <TypeDescriptor TypeName="System.Data.IDataReader, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" IsCollection="true" Name="Reader">
          <TypeDescriptors>
            <TypeDescriptor TypeName="System.Data.IDataRecord, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="Record">
              <TypeDescriptors>
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" IdentifierName="pid" Name="pid" />
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="bldtyp" />
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="ethno" />
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="sx" />
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="cgp" />
              </TypeDescriptors>
            </TypeDescriptor>
          </TypeDescriptors>
        </TypeDescriptor>
      </Parameter>
    </Parameters>
```

Method Instances

This is a description of the method types used across the described dataset and are referenced by the ADO.NET TypeDescriptors used in the return parameters. The first method is of type “Finder” which allows access, via the filter descriptor, to the whole table. The second is of type “SpecificFinder” which returns a single record via the identifier described in the “Entity” node. The Finder method instance is required for the profile import mechanism.

```
                    <MethodInstances>
      <MethodInstance Type="Finder" ReturnParameterName="@profilesupp" ReturnTypeDescriptorName="Reader" ReturnTypeDescriptorLevel="0" Name="Find_profilesupps_Instance" />
      <MethodInstance Type="SpecificFinder" ReturnParameterName="@profilesupp" ReturnTypeDescriptorName="Reader" ReturnTypeDescriptorLevel="0" Name="Find_profilesupp_Instance" />
    </MethodInstances>
  </Method>
```

Index Method

This method allows the indexer to crawl the BDC for use in the search engine. It uses the same principles described earlier with the exception of the method Instance which is of type “IDEnumerator”, the IDEnumerator instance returns data in a format that is understood by the crawler.

```
  <Method Name="FindAll_profilesupp">
    <Properties>
      <Property Name="RdbCommandType" Type="System.Data.CommandType, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089">Text</Property>
      <Property Name="RdbCommandText" Type="System.String">Select "pid" from  profilesupp</Property>
    </Properties>
    <Parameters>
      <Parameter Direction="Return" Name="@profilesupp">
```

```
        <TypeDescriptor TypeName="System.Data.IDataReader, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" IsCollection="true" Name="Reader">
          <TypeDescriptors>
            <TypeDescriptor TypeName="System.Data.IDataRecord, System.Data, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" Name="Record">
              <TypeDescriptors>
                <TypeDescriptor TypeName="System.String, mscorlib, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" IdentifierName="pid" Name="pid" />
              </TypeDescriptors>
            </TypeDescriptor>
          </TypeDescriptors>
        </TypeDescriptor>
      </Parameter>
    </Parameters>
    <MethodInstances>
      <MethodInstance Type="IdEnumerator" ReturnParameterName="@profilesupp" ReturnTypeDescriptorName="Reader" ReturnTypeDescriptorLevel="0" Name="FindAll_profilesupp_Instance" />
    </MethodInstances>
  </Method>
 </Methods>
</Entity>
</Entities>
</LobSystem>
```

Appendix B

Microsoft Office SharePoint Server 2007

B.1 Component Topologies

Each component of the system can be located on a separate server or, for smaller environments, located on the same server. This includes the Database.

B.1.1 SharePoint Farm

A collection of these components running in any topological configuration is known as a farm. The diagram below represents a typical MOSS medium farm for a medium sized application of < 30,000 users.

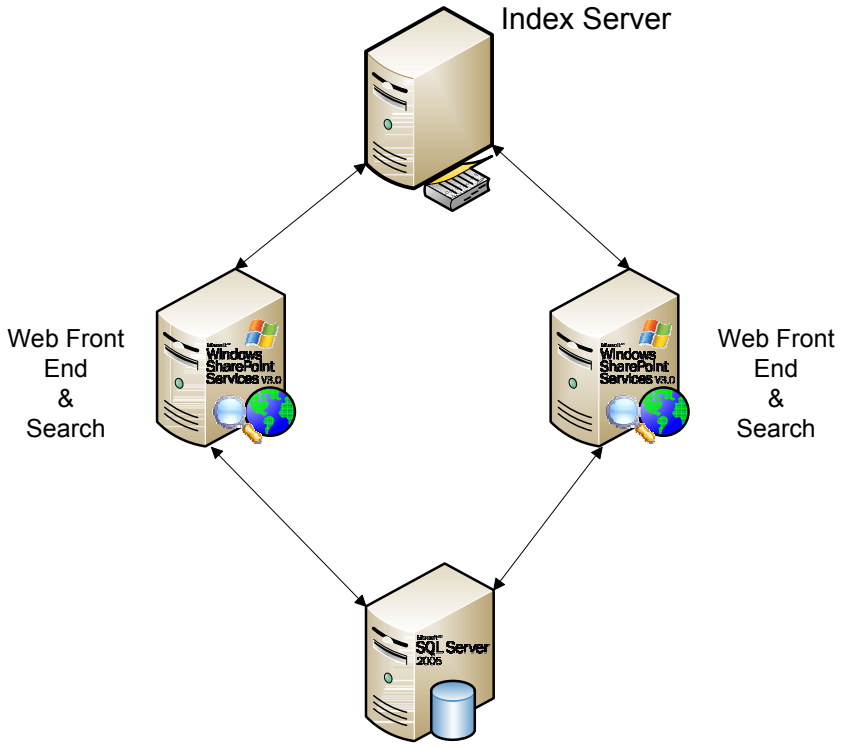


Figure B 1 Example MOSS farm topology

.

B.2 Windows SharePoint Services (WSS) 3.0 sites

MOSS is built under Windows SharePoint Services 3.0 (WSS 3.0). WSS 3.0 is a collaborative web technology known as web 2.0. An expression of WSS 3.0 is the front end component which is basically a web 2.0 enabled web site known as a WSS 3.0 site. MOSS's structure is made up entirely of WSS 3.0 sites. A site can be configured then saved as a template with or without any inclusive data.

Features of WSS 3.0 site

- User groups
  - based on AD security principles
- User roles
  - predefined set of Roles granting different levels of access to elements in the site. New roles can defined and custom access levels can be created
- Lists
  - A list is a container in which items of information can be stored shared, managed and secured; lists are fundamental to WSS 3.0 sites. Just as MOSS is a structured collection of WSS 3.0 sites, WSS 3.0 sites are made up entirely of lists, so, a WSS 3.0 site is a structured collection of lists. A list can also be considered as a table in a role-based security enabled database.

    - List Features
      - Require approval
      - Integrate e-mail with a list
      - Customize permissions
      - Create and manage views
      - Use formulas and calculated values
      - Keep informed about changes
      - Share list information with a database program
      - Use lists consistently across sites
      - Email integration

    - Types of list
      - Announcements
      - Contacts
      - Discussion boards
      - Links
      - Calendar
      - Tasks
      - Project Tasks
      - Issue Tracking
      - Survey
      - Custom

- Document Library
  - Is a list that contains files as well as information about those files, it has the same features as a list

B.3 WSS 3.0 Site Templates

A WSS 3.0 site can take on many roles. When a site is created a template is applied to the underlying WSS technology to produce different functionality for different uses. A template merely manipulates the basic list technology to produce functionality. Templates can be created by developers to produce custom sites.

Some examples of site templates are:

- Meeting workspace
- Document workspace
- Publishing sites
- Team sites
- Blog
- Wiki
- News
- Reporting

Display and Web Parts

WSS 3.0 sites and pages are rendered as .aspx pages via the Microsoft ASP.NET 2.0 web server technology. The display of a site is controlled by asp web user interface (UI) controls and .NET 2.0 Web parts which are also an extension of .NET UI controls. A page or site can be edited using a rebranded version of Microsoft Front Page called SharePoint Designer. Using this tool means that pages can be customised to suit the branding needs of the organisation. A Web Part is a floating, collapsible panel of information. A user, when editing a page, can drag and drop these web parts onto Web Part zones excluding the top and left navigation areas of the page.

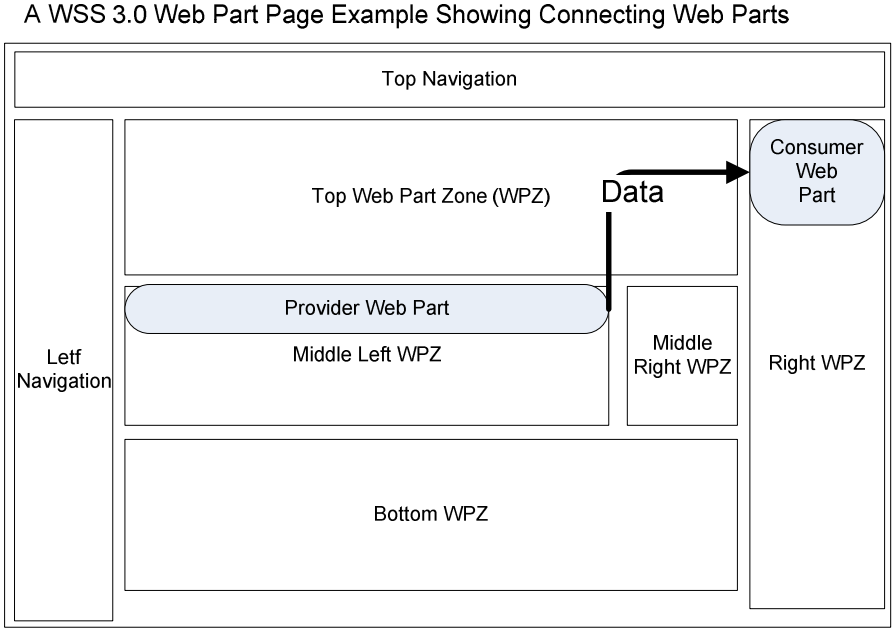


Figure B 2 Web part page showing connected web parts

Web Parts can be used to express any information from within a MOSS farm. They can also be used to display data, external web pages and email components. Web Parts are fully programmable, developers can create any piece programming and express it through a web part. In addition, if the programming code is included, web parts can be connected, provider web parts send an item of information to consumer web parts when the page is loaded forming run time relationships between information components. Ajax controls via .NET 3.5 libraries can also be implemented in connecting web parts, negating the need for page refreshes.

B.4 Site Collections

A site collection is a WSS 3.0 site known as a Top Level Site (TLS) with other WSS 3.0 sites hanging from it known as Webs. The structure of an organisation can be mapped to structure inside a portal. The following diagram depicts just one way this could be achieved in the Hospital example.

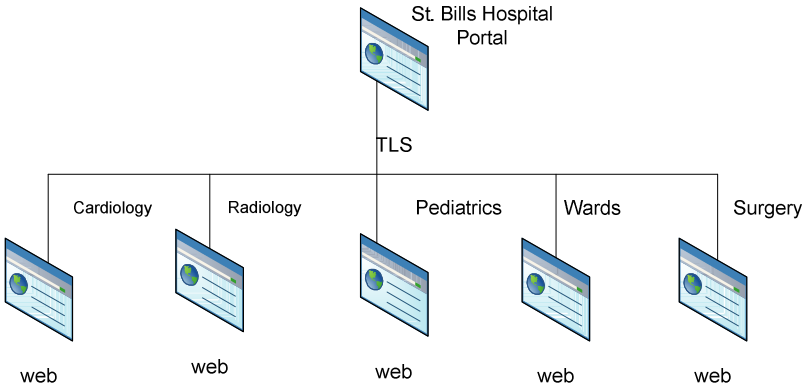


Figure B 3 Example site structure

Using “Wards” in the above diagram as an example and drilling down into that branch of the Hospital we might find the following example.

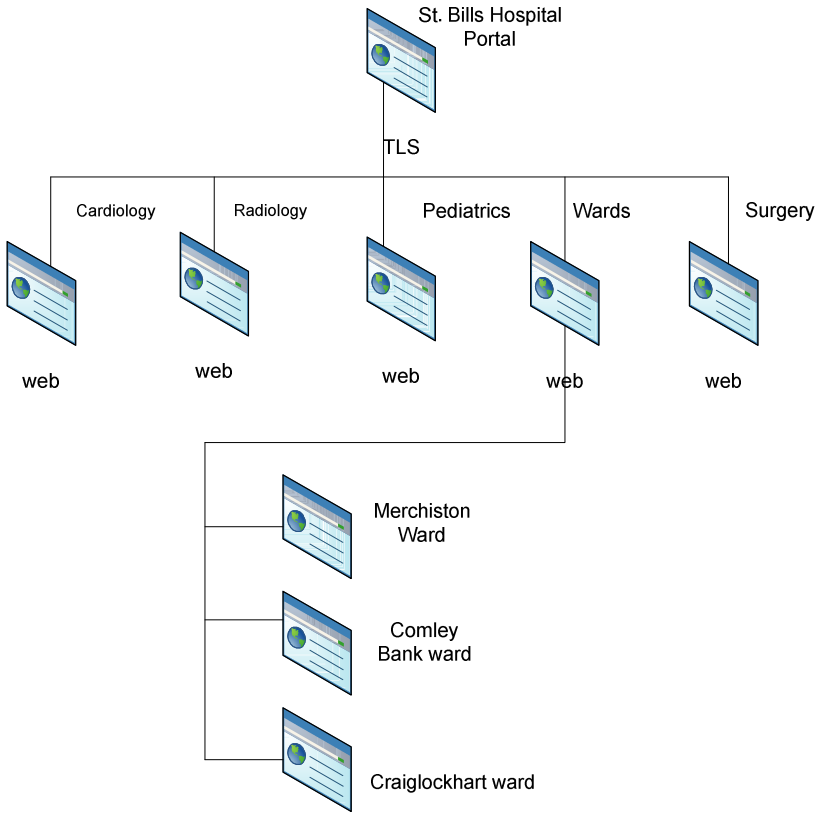


Figure B 4 Exmaple sub-site structure

Each ward WSS 3.0 site contains all the features described earlier in this document. Depending on how the information is required, a ward site may contain lists which in turn may contain information such as bed status, patient list, staff lists, hygiene rotors and chart data to name but a mere few. Other departments may contain similar information in terms of staff activity but not other information such as blood test results from laboratories.

B.5 WSS 3.0 Roles and Permissions and Memberships

It is important to discuss the differences between Roles, Permissions and Memberships in this framework when defining the security model for a given environment. SharePoint has a substantial set of permissions allowing various operations on a WSS 3.0 site. Collections of these permissions make up permission levels which in turn can applied to individuals or groups. The permission set is accessible to framework administrators and developers allowing new permission levels to be created based on customised permission sets or simply to name permission levels in accordance with organisational needs.

B.5.1 Default Permissions

The default permission levels in relation to each respective WSS 3.0 are:

- Full Control – Has full control over all aspects of the WSS 3.0 site
- Design – Can view, add, update, delete, approve and customise WSS 3.0 site content

- Manage Hierarchy – Can create and edit WSS 3.0 sites, pages, lists and documents
- Approve – Can edit and approve pages and content
- Contribute – Can add items to lists and participate in Wiki sites and Discussion
- Read - can view list items and site pages
- Restricted Read – Can only read current version of information contained within a site
- View Only – this is similar to read except that document that can be view via a server side operation can only be viewed via that operation, i.e. if a server has a document viewer the user cannot use a client side program such as Microsoft Word to open it. This ensures documents stay server side and are not duplicated on client machines.

B.5.2 Site Groups and Roles

When a WSS 3.0 site is created groups are created based on both the chosen template and some of the permission levels defined above, each group can contain individual users or AD security groups. The name given to these groups give rise to the intend purpose of the group containers which is to allocate roles. When a user's membership of a site is defined by these groups, it the membership of the site group defines the users role which is itself governed by the underlying permission level given to the group. Again, custom groups can be created to suit security needs or to comply with organisational naming conventions. When a Top Level Site (TLS) collection is created in a MOSS environment using a collaboration portal template, the default groups supplied are:

- Owners – (usually preceded by the title of the WSS 3.0 site) – Full Control
- Designers - Design
- Hierarchy Managers - Manage Hierarchy
- Approvers – Approve
- Members - (usually preceded by the title of the WSS 3.0 site) – Contribute
- Visitors - (usually preceded by the title of the WSS 3.0 site) – Read
- Restricted Readers – Restricted Read
- Viewers – view only

Groups that also added which are supplemental to the previous discussion are:

- Quick Deploy Users – This group is for operations specific to rapid publishing
- Style Resource Readers – This has permission to administrative galleries the restrict access to master pages and styling libraries

A permission level can be applied to any level of a WSS 3.0 site. As previously stated, WSS 3.0 sites are made entirely from lists which are, in turn made up entirely of list items. Each list item can have none or more permission levels applied to it via group or individual assignation, making it possible to secure every element within the WSS 3.0 environment.

B.5.3 Permission Inheritance

Each WSS 3.0 site that is created from another WSS 3.0 site, i.e. a child site, has the option to inherit permissions from the parent site. In this case, groups are not created as all authorisation requests refer to the parent site group permissions. Permission inheritance can be broken or re-established at any level and at any time.

B.6 Content Type and Site Columns

Content types are the core information types that make up the framework. An example of a content type is a “Basic Page”, this content type used by the system display .NET aspx pages. Content types can be based on another content type when created, the Basic Page example described earlier is based on content type “Document” this is known as the parent content type. “Web Part Page” is a content type based on the parent type “Basic Page”. It is possible create new content types based on any of the default content types available, newly created content types can also be used as parents for additionally created types. Site columns allow global use of list columns across the framework. For example; if a custom list is created with a column “PatientID”, PatientID may be required for use in another list, Instead of creating an additional column called PatientID for the additional a site column is created and chosen from a list of site columns, site columns are based on content types.

B.7 Workflow

As stated in section (sec), workflow allows business processes to be mapped into electronic format, the .NET 3.0 and, optionally, 3.5 framework is required to run workflow in MOSS although it is noted that .NET 3.0 workflow component can run in any host application such as a DICOM application or HL7 compliant applications, Figure B 5 illustrates the .NET 3.0 workflow architecture based on a MOSS host application. Workflow consists of several class libraries that allow tasks to be created, manipulated and monitored over a time line (See Figure B 5).

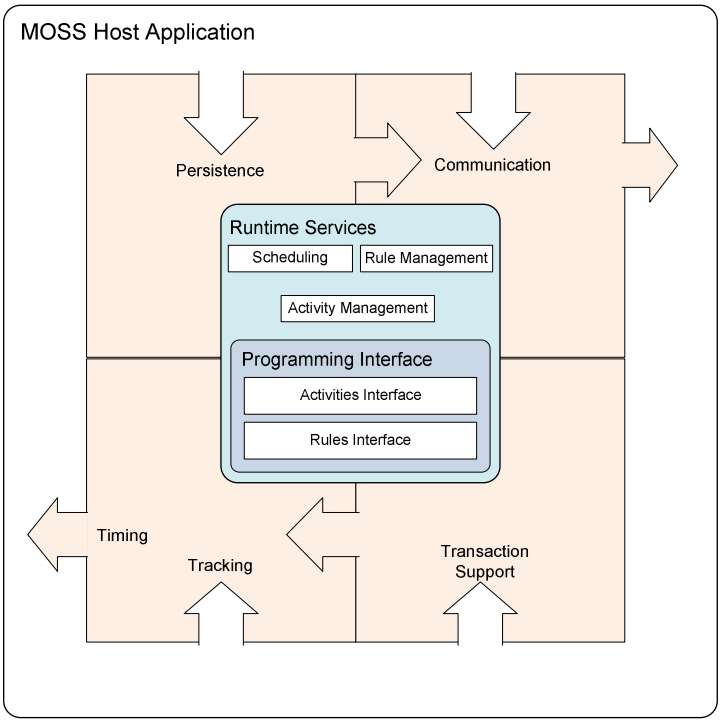


Figure B 5 Workflow architecture

B.7.1 Standard SharePoint Workflow

There are several standard workflow types available without the required programming knowledge. One example of a standard MOSS workflow is called an approval workflow, this can be created by users, and it basically provides a platform for users to notify other users that they need to check something before routing onto the next stage. When complete the approver will change the status of the task to "approved" and something else happens, what that something is, is defined by the business process itself. There are essentially two types of .NET 3.0 workflow: sequential, this means the work has a beginning and an end, or, state based, this is when the workflow does not end as such, it just exists and changes its state or it doesn't exist. A state based workflow closely follows the principles of Finite State Machines and is created using similar design templates. Workflows can be customised to perform any programmatic task either native or third party. A workflow can start another workflow synchronously or asynchronously, this gives rise to the possibilities of the ability to derive highly complex business process and express them digitally. A workflow process can span years, which can equate to the length of time some healthcare cases can take. In context of a patient-centric system, workflow could be a key aspect of the attempt to challenge the issues discussed. There are several business processes that surround the patient, the patient's arrival at a hospital to the patient's departure could be considered as one large linear workflow. All the processes therein could then be considered as atomic workflows;

B.7.2 Workflow Example

If a blood test is required for any patient then a blood test workflow could be programmed, this may involve a doctor requiring blood samples and nursing staff initiating a workflow. Blood taking nurses would see the entry and, if necessary, porters may see the entry to. At this point some action is taken, i.e. an appointment for blood to be taken is made and the porters are informed where when to move a bed ridden patient or, the blood taking nurse visits the patient and takes blood. The workflow can them move into a new state, for example the blood would then be stored awaiting transit to the laboratory. The details of the stored blood could then be displayed on the screen of those transporting the blood. This could prevent mistakes in transit or left behind samples, once those transporting the blood have carried out their task the workflow moves to a new state, and the laboratory examine the sample and collect results. Once complete the results are made available via a link to the original requester, who is informed when the laboratory moves the state of the workflow into the published state. The workflow could be ended at this point, but to ensue the information has been viewed it would probably be more efficient to have the original requester trigger the termination of the workflow by confirming receipt of the results.

From the description of the example earlier, we can see that there is a definite business process surrounding the activity of getting a patient's blood and it would appear as is that a single workflow would be ideal for that task. To further increase flexibility, each activity in the process could also be a separate atomic workflow, which could be started by each other;

B.7.3 Atomic Workflow Example

The porter's job is to move people around the hospital, ensuring that patients, who can't help themselves, get to the right place to receive the treatment, analysis or care.
It is reasonable to assume that their tasks do not centre around blood tests, therefore it may be more efficient to treat the activity of moving patients as separate workflow that can started by other workflows not necessarily to do with sampling blood.

B.7.4 Bed Model

A bed continually exists with the hospital and only changes its state, the bed can be expressed in state based workflow. In additional to the basic states of occupied and unoccupied, the bed also required tasks to be actioned upon it, i.e. cleaning and maintenance, or requiring cleaning, requiring maintenance. The beds state could be changed by a number of staff members. For example:

- The ward Nurse
  - The ward nurse and other nurses could change the bed status to requiring cleaning or requiring maintenance
  - The ward nurse could change the bed status empty after some defined protocols are observed
- Hospital Admissions
  - Hospital admission could change the bed's status to full
- Hospital Discharges
  - Staff here could change the beds status to empty
- Hospital Morgue
  - Unpleasant as it is, staff here could also change the bed status to empty

B.7.5 Patient-centric Model

In a patient-centric system, the information about the patient should be paramount when defining business processes and implementing workflow. This requires analysis of patients movements through a health care system, the systems required to provide HCPs with the correct information and the HCPs activities that the patient may require. The Bed Model, described earlier, may be viewed as "Bed Centric" as the only interest, is in the state of the bed; this is fine for those who only require knowing the bed's status to complete their tasks. In terms of patient care, the model is similar in nature but far more complex in detail, the basis of knowing the status of a patient is similar to the bed model, but the information required to assess the patient's status differs depending on the HCPs tasks for that patient. The technology discussed will provide this functionality through publishing, disparate data consumption, targeting and workflow. Information maintained by disparate systems will still be maintained by disparate system owners, consumed by HCPs when required for the patient, information is static, residing in the same place, rather than moving information physically between HCPs, as is the case at present, references to the correct information should only be moved, this means the information cannot get lost, only the reference to it can, leaving the information itself in a well known place.

B.8 Key Performance Indicators (KPI)

KPIs have their roots in the business world and are regularly used as data for sound bites in presenting performance figures in an organisation. KPIs are usually analogues for some aspect of information; they are sometimes represented by some rudimentary graphic such as a traffic light or speedometer. If we are to apply this notion to the Bed Model described earlier, we could represent the bed status as green for empty, yellow for requiring cleaning or maintenance, or red for occupied. Provided that the state of the bed is moved by the respective completions and indeed input from the staff members, an accurate picture of the state of beds in wards, floors, hospitals and even other hospitals could be viewed by the hospital admissions staff that require it. Representing this data through KPI can give an "at-a-glance" service not only for the local system but other hospital systems as well.

B.9 Dash Boards

A dash board "A visualization of important information, often tailored to a specific role or point of view, consolidated and arranged on a single screen"
www.pilotsoftware.com/resources/pm_glossary.html
It is clear, from this definition of the word dashboard, that to effectively tackle the majority of issues described in the main text, that some kind of implementation of dashboards is required. Since MOSS allows the creation of pages, the addition of web parts containing information on those pages; allows that information to be native, foreign and custom and allows that information to be secured in a role-based way, there exists the ability to create dashboards, in fact, Microsoft used the term "dashboard" when describing this functionality in Version 1.0 of the product. In terms of all the examples described, the link that is missing is how the HCPs are to get information from and interact with the system. Dashboards tuned to an HCPs role by targeting the correct information on the patient will allow information to be directed efficiently, workflow to ensure the HCP's are dealing with requests in a timely fashion and arrive at the right HCP, at the right time with a reference to the right information will increase the efficiency of the flow of information though the system and therefore, hopefully, support the hypothesis.

B.10 Shared Services

Shared Services are set of features of MOSS that exist as a separate modular entity known as Shared Service Providers (SSP); the services this entity includes are:

- Business Data Catalogue
- Search Service
- Audiences
- User Profiles
- Personalisation Sites
- My Sites
- Excel Calculation Services
- Usage Reporting

B.11 Business Data Catalogue (BDC)

The business data catalogue is part of MOSS shared services and is used to consume data from disparate sources, simply writing a ODBC connection to any data and consuming data directly seems more efficient and is certainly possible, but, the data is then static in the environment, it only means something specific to that requirement at that time. What the BDC does is describe the data as a hierarchical structure or schema allowing the framework to treat the data as if it were part of the MOSS data structure and as so then run methods over the returned BDC data as with a native list. This means the data is reusable in the entire information infrastructure. The BDC can use several types of access techniques found in most databases. Role-based authentication can be used or a single privileged user can be used in a built in single sign on module comprising of an encrypted database storing user credentials.

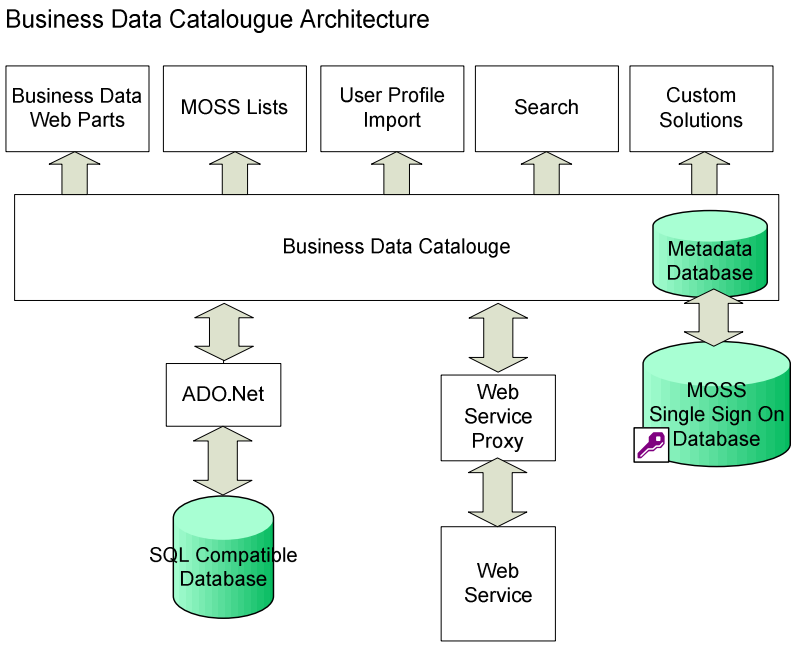


Figure B 6 BDC Architecture

B.12 Metadata Container Model

The BDC uses XML to describe the aspects of the data gathering process. A file known as an Application definition File (ADF) containing the XML is loaded into a part of the BDC known as the Application Registry. When creating an ADF file, the XML used must be valid and correspond to the Metadata container model which is the Object Model used by the BDC to navigate through a described hierarchy.

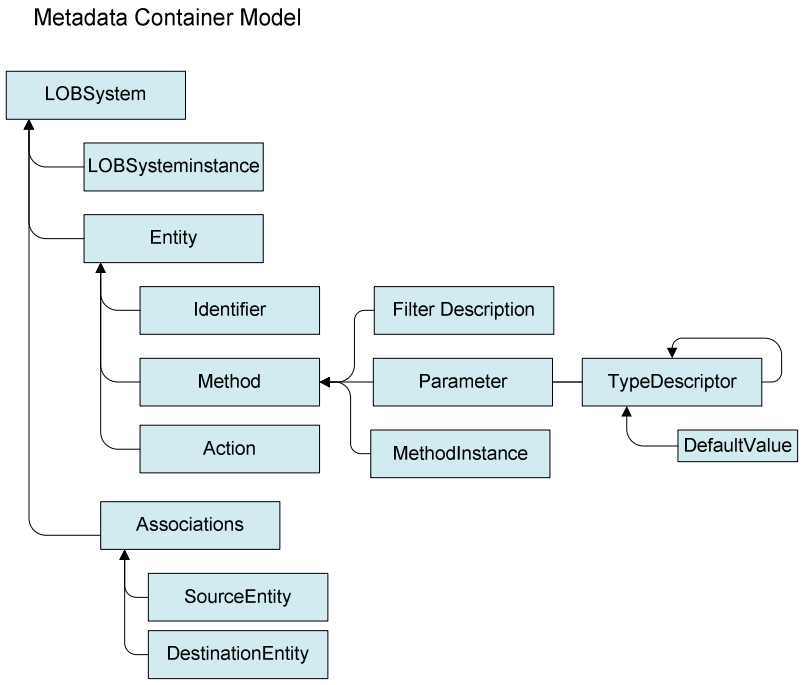


Figure B 7 Metadata container model

The root Node of the ADF file is the LOBSystem (Line of Business). Child nodes include:

- LOBSysteminstance – This contains information about the data connection and any Single Sign On (SSO) providers used
- Entities – Information on the disparate source, as well as all the methods identifiers and actions associated with that source
- Associations – Contains location information for connecting entities

A number of methods can be used to add extra functionality to the data retrieved from the disparate source, these include index methods which allows the data to be crawled by the index server and methods that allow some interaction with the data. BDC data can be expressed in lists and web parts, associative methods can be added to ADFs which allows BDC data to be passed between web parts at runtime.

BDC data is available in a one way fashion, i.e. no write backs, this means that data is only consumed through the MOSS infrastructure and any interaction is on the parent data set. The BDC also interacts with the profile database via the indexer, this means that extra attributes from disparate sources can be defined and incorporated into user profiles see Section 4.7.

B.13 Search Service and Indexing

One of MOSS's components is an Index server. Indexing works by using an index engine to navigate through content sources, gathering Meta data and storing it in a file known as an index volume, this gathering process is known as crawling. Users use the query engine to inter-

rogate the index volume and present results. In addition to crawling MOSS sites, the index engine is capable of crawling other content types such as a Network File Systems and external web sites.

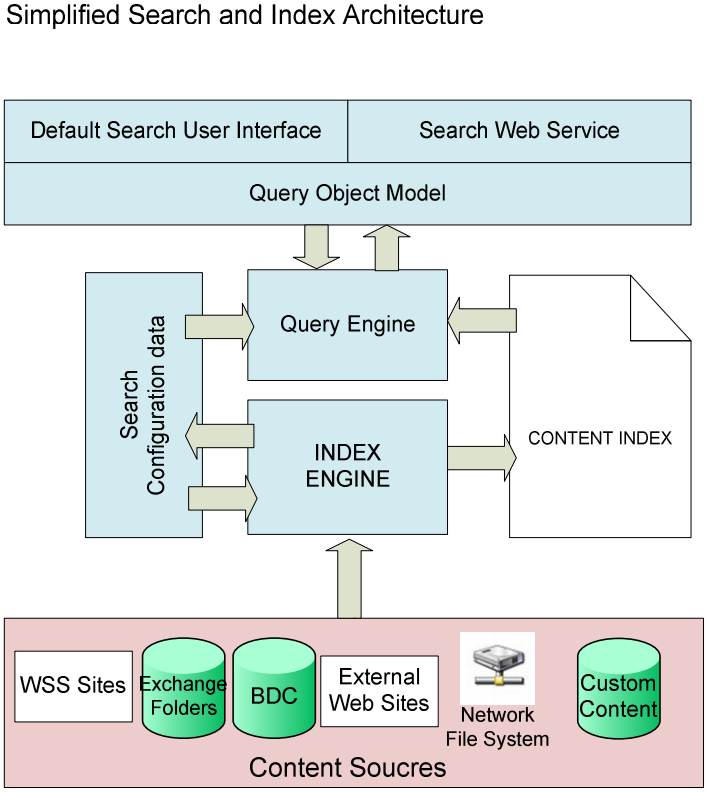


Figure B 8 Search and index architecture

Crawler rules can applied to the index engine which governs the extent to which a source is crawled and which aspect, if any, of the source is to be filtered out. Indexing is configured using the search service in a SSP, content sources crawl logs and rules along with search query configurations such as key words scopes and best bets are all stored in a configuration database.

B.13.1 Business Data and Indexing

Using the business data catalogue feature in MOSS with the indexer means that data from disparate sources such as a Local GP MS Access database or proprietary laboratory databases can also be crawled, again extending the amount of information available to HCPs, this could help in diagnostic decisions.

B.14 Audiences

Audience configuration allows information to be directed at groups of users, these groups can be defined by:

- SharePoint groups

- Active Directory groups
- Active Directory distributions lists
- Any user attribute of the profile database
- Any user attribute of Active Directory
- Any combination of the above

One example would be for an audience to be created based on the Skills attribute in the profile store, with a field value of Surgeon, this means that a piece of information or a list that is audience targeted to this audience will only be seen by surgeons. One point to make about audiences is that it appears to imply security, but that is not the case, audience targeting merely governs the information that is displayed; this does not stop a user from accessing the list item directly. This works well in a medical environment as HCPs, whilst not always requiring to see some information about a patient, should not be prevented from accessing it if required.

B.15 Profiles

Active Directory is a structured schema defining attributes of a user. “Preferred Name”, Last Name” and “Telephone Number” are some examples of these attributes. MOSS populates its user base by importing profiles from Active Directory and storing them in a separate database provided by an SSP, known as the profile store. MOSS contains additional profile properties that can be associated with a user as well the facility to create new attributes of a user and associate them with addition information sources via a Shared Service Provider’s Business Data Catalogue feature. A user must be present in the profile store to utilise the services provided by an SSP. A user does not need to be present in the profile store to gain access to a WSS 3.0 site; this is mechanism which allows WSS 3.0 sites to run independently of MOSS. The profile store is filled via schedulable process involving LDAP queries to a Domain Controller (DC) holding the AD component. The returned user account are then written into the profile store located on the SQL Server. The search service and indexer are used to perform profile related queries in a SSP.
Example
A large farm could be installed into a regional hospital, the sites that hang off the hospital site could be the local hospitals and the sites that hang off them could be GPs and Medical centres, in this scenario there is only requirement of a single SSP, this means that the security and access rights to information in the entire area is covered, allowing an HCP to operate from any of those given sites without further action. This principle clearly outlines several advantages in terms of centralised services but does not take into account the communication link that exists between these sites, if the link goes down there is no way to operate.

B.16 Multiple SSP and Search

Multiple shared services provide a way to isolate search results; this may be from legal requirement to keep records from being viewed by unauthorised personnel whilst retaining some statistical functionality. In the context of patient care, a patient may be entitled to a vast amount of information from a farm, general information that everyone is entitled to view and

personal information under FOI, HCPs may be entitled to compare patient data for statistical analysis, this information may not be viewed by patients as it may contain data from other patients. In any case, there is need to prevent one from seeing the information in the other. Using separate Shared Services allows the access account and crawl rules to be different in for every shared service provider, this means that some information will not be available through search to prevented users. Limitations in configuration of multiple Shared Service Providers (SSP) mean that the development of the MOSS framework is required to achieve exact functionality.

Example

A WSS 3.0 site structure AKA portal AKA Microsoft Internet Information Services web application can only be associated with one SSP. This means that using the default search API will only return results from that one SSP, to get search results from any other SSP the code must be customised to include the alternative indexes in any search queries.

B.17 MOSS Dependencies

In addition hardware, a MOSS installation also requires:

- Microsoft Windows Server 2003 or 2008
- Microsoft Internet Information Services (IIS) 6.0 or later
- Active Directory (Domain Controller)
- SQL Server 2000/2005 or 2008
- .NET framework 3.0
- .NET framework 3.5

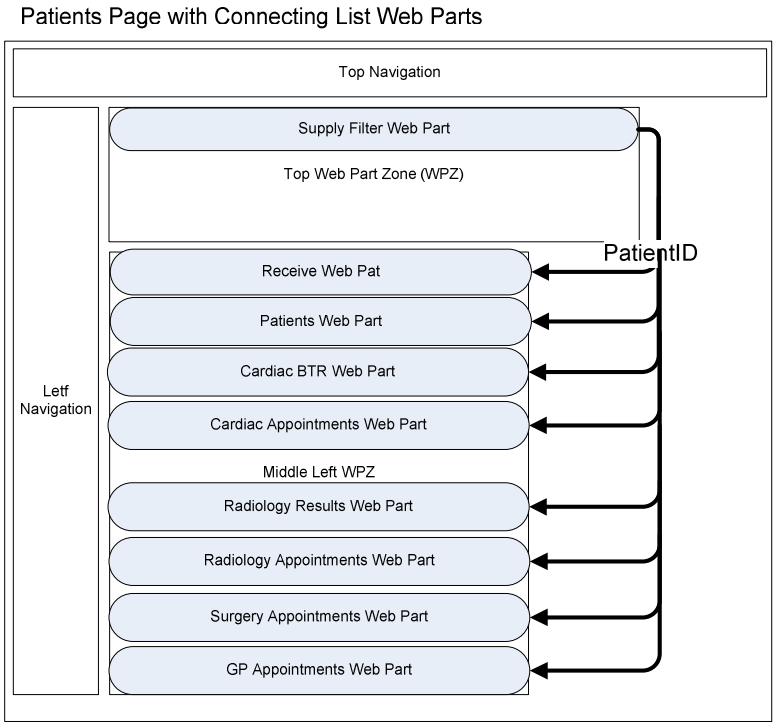
Appendix C

C.1 Design Deviation flat list approach

This approach was achieved by recreating all of the departmental lists created Section 5.5.1, inside the TLS, the patientlookup.aspx page created in Section 5.5.3 was used again using the original design from Section 4.9. MOSS audience targeting described in Appendix B.14 was then added to the web parts based on a similar access matrix as used in Section 5.5.2.

C.2 Patientlookup.aspx Page Design

The Patientlookup.aspx page created in Section 5.5.3 is edited again. The list web parts representing each departmental list are taken from the TLS's web part gallery and placed on the patientlookup.aspx web part page as shown in Fugure C 1.



Fugure C 1 Patientlookpage design

C.3 Audience Targeting Configuration and Testing

Verification that information was being displayed based on the "Supply" filter text variable was carried out and audience targeting was then applied by setting each web part to be filtered based on site group membership (See Section 4.8). The information divide in this context was between department Administrators and Medical Staff based on premise that medical staff may not need to see the appointment information from another department but administrators may require this to manage patients, blood test results information could be seen by GP administrators and all medical staff whereas radiology results are usually interpreted by HCPs so there was need for Administrators to see this information, results from Cardiology blood tests could not be seen by Radiology and vice versa. Table C 1shows site

group audiences in the columns and the lists are defined in the rows, the ‘x’ represents which audience was allowed to see which list.

Table C 1 Audience Tables

	Radiology Owners	Cardiology Owners	GP Administrators	GP Doctors	Surgery Surgeons	Surgery Administrator
Cardiac BTR		x	x	x	x	
Cardiac Appointments		x	x			x
Radiology Results	x			x	x	
Radiology Appointments	x		x			x
Surgery Appointments			x		x	x
GP Appointments			x	x		

C.4 Verification

Each web part had an audience applied corresponding the columns in Table C 1. The patient-lookup.aspx was then returned a normal operating state. The page was then visited, using each HCP user account and results recorded.

Table C 2 GP View

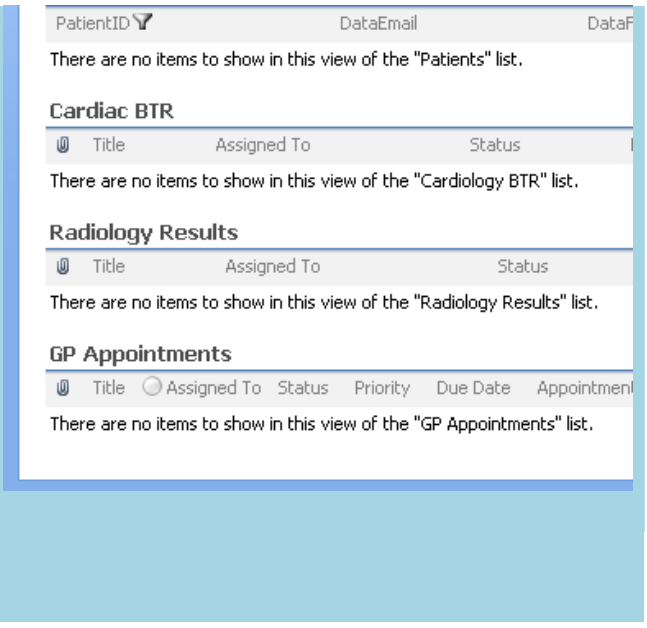
Office\GP	GP Doctors	Screen Shot
Cardiac BTR	x	
Cardiac Appointments		
Radiology Results	x	
Radiology Appointments		
Surgery Appointments		
GP Appointments	x	

Table C 3 GPSecretary view

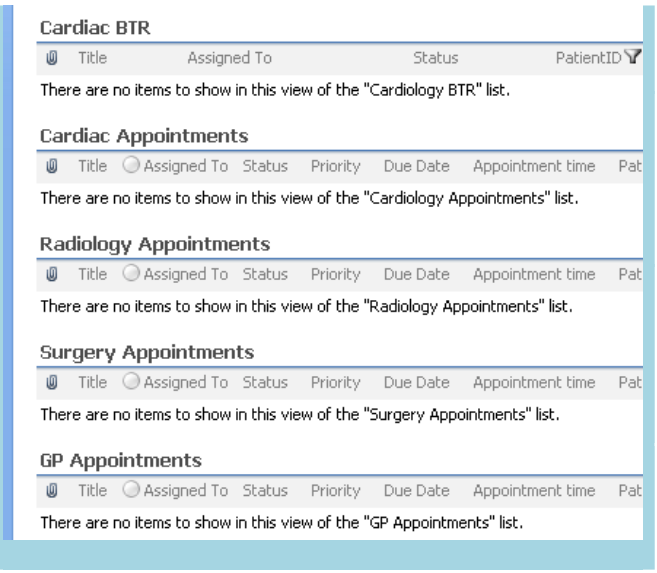
Office\GPSecretary	GP Administrators	Screen Shot
Cardiac BTR	x	
Cardiac Appointments	x	
Radiology Results		
Radiology Appointments	x	
Surgery Appointments	x	
GP Appointments	x	

Table C 4 SurgeonSecretary view

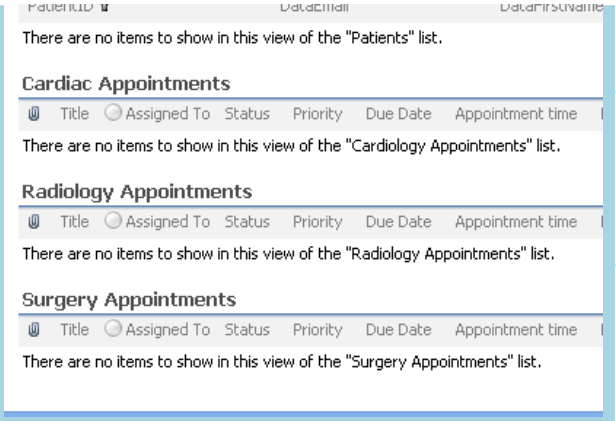
Office\ SurgeonSecretary	Surgery Administrator	Screen Shot
Cardiac BTR		
Cardiac Appointments	x	
Radiology Results		
Radiology Appointments	x	
Surgery Appointments	x	
GP Appointments		

Table C 5 Surgeon view

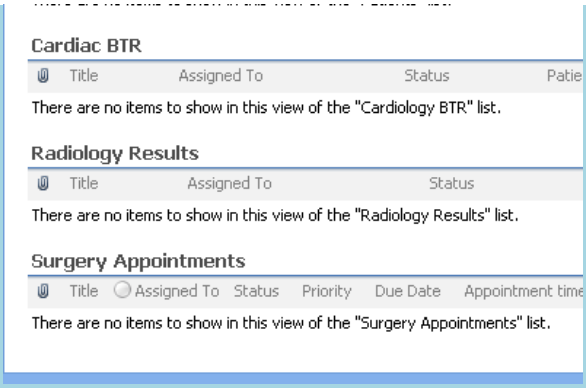
Office\ Surgeon	Surgery Surgeons	Screen Shot
Cardiac BTR	x	
Cardiac Appointments		
Radiology Results	x	
Radiology Appointments		
Surgery Appointments	x	
GP Appointments		

Table C 6 CardioLab view

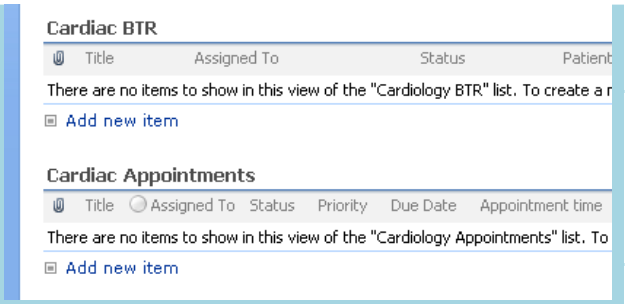
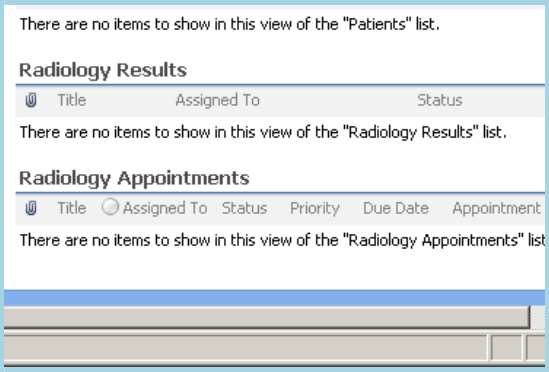
Office\ CardioLab	Cardiology Owners	Screen Shot
Cardiac BTR	x	
Cardiac Appointments	x	
Radiology Results		
Radiology Appointments		
Surgery Appointments		
GP Appointments		

Table C 7 RadioLab user view

Office\ RadioLab	Radiology Owners	Screen Shot
Cardiac BTR		
Cardiac Appointments		
Radiology Results	x	
Radiology Appointments	x	
Surgery Appointments		
GP Appointments		

When the PatientID is added to the ”supply” text box on the PatierntLoookup.aspx page, the information is received by all the web parts Fugure C 1, however, if the web has been targeted it will only display to those in the defined audience.
It was noted that the view of the lists the web parts were looking could be changed. One example of an application of this change would to create a view that looks at the dates of appointments allowing trimming, via filters in the view, of old data. It was also noted the HCPs did not have permission to see the disparate source; this was because the privileged user for the source was set to OFFICE\Domain Admin.

Appendix D

D.1 States and Methods

This section lists tables showing each state and the activity names, activity types (sec) and any overridden methods contained within.

Table D 1 Workflow initiation state activities

Workflow1InitialState			
Activity Name	**Activity Type**	**Method Used**	**Description**
eventDrivenActivity1	Event Driven Activity	None	Mechanism for state machine workflow may contain child activities
onWorklowActiviated1	workflow activity		Instantiates the workflow
setStateActvitity1	Set State		This activity moves the state to the Set Task (fig) state.

Table D 2 Set Task activities

Set Task			
Activity Name	**Activity Type**	**Method Used**	**Description**
stateInitializationActivity1	State initialisation	None	Mechanism for state machine workflow may contain child activities
createTask11	Task	createTask1_MethodInvoking	Creates a workflow task list item
setStateActvitity4	Set State		This activity moves the workflow to the waitForDepartment (fig) state.

Table D 3 Waiting for department state activities

Waiting for Department			
Activity Name	**Activity Type**	**Method Used**	**Description**
stateInitializationActivity3	State initialisation	None	Mechanism for state machine workflow may contain child activities
onGetStateUpdate	Task	onGetStateUpdate_MethodInvoking	Assigns the task to the departmental secretary when the state is entered
onDepTaskChange	Event Driven	None	Mechanism for listening to task change events, usually contains child

			activities
onStatusToHCP	Task	None	Persists properties of a task before and after it is modified.
isDepCancelled	If Else	None	Routes the workflow based on the status property of the workflow task after the task has been modified. If the status is set to cancelled, the work flow terminates. If not, the workflow moves into the waiting for department state(this means that an error state can be achieved)
setListCancelled	Code	setListCancelled_ExecuteCode	Set the surgery referrals ltem status to cancelled
setStateDepCancelled	State Transition	None	Move the workflow into the testState state
setAppointmentStatus	Code	setAppointmentStatus_ExecuteCode	Set the surgery referrals ltem status to Waiting For HCP state
setWHCP	State Transition	None	Move the workflow into the waitingForHCP state

Table D 4 Waiting on HCP state activities

waitingForHCP			
Activity Name	**Activity Type**	**Method Used**	**Description**
stateInitializationActivity2	State initialisation	None	Mechanism for state machine workflow may contain child activities
setAssignedTo	Task	UpdateTask	This activity assigns the task to the HCP, populates the HCPUID column so that when the task changes the HCP is recorded as the attending for this appointment, increases the due date by 2 days and changes the title of the task.
sendEmail1	Send Email	sendEmail1_MethodInvoking	Sends an mail outlining the contents of the task to the HCP to which the task is assigned
eventDrivenActivity3	Event Driven	None	Mechanism for listening to task change events, usually contains child activities
onHCPAction	Task	None	Persists properties of a task before and after it is modified.
checkErrorEvent	If Else	None	Error condition for the HCP action, if the status of the task does not equal HCP Accept or HCP

			Reject, the state resets and no action is taken
status1	If Else	None	Check task status after change is made in onHCPAction. If the task's status is HCP Accept route the workflow down the true branch
setListStatusWFA	Code	setList-StatusWFA_ExecuteCode	If Status1 is true, this code sets the surgery referrals's status field to waiting for appointment
sendAcceptenceToPatientEmail	Send Email	sendAccep-tenceToPa-tientEmail_MethodInvoking	Sends an Email to the patient containg appointment information
setStateWFA	State Transition	None	Moves the workflow into the waitForAppointment state
isHCPReject	If Else	None	Condition to see if the task's status is set to HCP Reject or not
setStateWFD	State Transition	None	Moves the workflow back into the waitForDepartment state
setListStatusWFD	Code	setListCycle	Change the surgery referrals item status field to reflect the task's state
setCode	Code	setListStatusPatien-tRej_ExecuteCode	Change the surgery referrals item status field to reflect the task's state
setCancelledHCP	Code	setListCancelled_Execute Code	Change the surgery referrals item status field to reflect the task's state
setEndStates	State Transistion	None	Move the workflow into a completed state

Table D 5 Waiting for appointment state activities

waitForAppoinment			
Activity Name	**Activity Type**	**Method Used**	**Description**
stateInitializationActivity4	State initialisation	None	Mechanism for state machine workflow may contain child activities
upDateTask1	Task	upDateTask1_MethodInvoking	This activity assigns the task to the Patient, adds the patient to the surgery site as a user and changes the due date for the

			task.
onAppointmentChange	Event Driven	None	Mechanism for listening to task change events, usually contains child activities
checkComplete	If Else	None	Condition to see check the complete status of the task. If the tasks status is complete the work flow completes
onPatientChange	Task	None	Persists properties of a task before and after it is modified.
ifElseActivity1	If Else	None	Error condition for the Patient action, if the status of the task does not equal Patient Accept or Patient Reject, the state resets and no action is taken
patientAction	If Else	None	Check task status after change is made in onPatientChange. If the task's status is Patient Accept, route the workflow down the true branch
updatePatientAccept	Task	updatePatientAccept_MethodInvoking	If patientAction is true, this code sets the surgery referrals's status field to patient Accept and changes the due date to that of the proposed appointment
updateCancelled	Task	CompleteTaskItem	See comleteTaskCode
comleteTaskCode	Code	CompleteTaskItem	When an appoint is either cancelled or complete this activity writes information to two lists and sets the surgery referrals status field to reflect either of these conditions
setPatientAccept	State Transition	None	Moves the workflow into the waitForAppointment state
setWaitingForDep setErrorPAT	State Transition	None	Moves the workflow into the waitForDepartment state
setErrorPatientAction setListStatusPatientRej	Code	setErrorPatientAction_ExecuteCode	Change the surgery referrals item status field to reflect the task's state
setTerminated	State Transistion	None	Move the workflow into a final testState

Appendix E

E.1 List Configuration Tables

Table E 1 Radiology results WSS list

Title	Assigned to	Status	PatientID	Date Tested	Results	HCPUID
Radology Results	Gerry Anderton	Completed	Patient01	6/25/2008 05:00 PM	Image	
Radology Results	Gerry Anderton	Completed	Patient01	6/26/2008 06:00 PM	Image	
Radology Results	Gerry Anderton	Completed	Patient03	5/7/2008 09:00 PM	Image	

Table E 2 Cardiology results WSS list

Title	Assigned to	Status	PatientID	WBC	RBC	Date Tested
Cardiology BTR	Jane Goodhart	Completed	Patient01	23.00	23.23	6/17/2008 09:00 AM
Cardiology BTR	Jane Goodhart	Completed	Patient02	123.00	430.00	5/7/2008 12:00 AM
Cardiology BTR	Jane Goodhart	Completed	Patient03	234.00	212.00	5/7/2008 12:00 AM
Cardiology BTR	Jane Goodhart	Completed	Patient01	232.00	121.00	5/7/2008 08:45 AM
Cardiology BTR	Jane Goodhart	Completed	Patient02	234.00	222.00	3/4/2008 03:00 PM

Table E 3 Radiology appointments WSS list

Title	Assigned to	Status	Priority	Due Date	Appointment time	PatientID	Location	Info About Appointment	Appointment Type
Appointment	Gerry Anderton	Not Started	(2) Normal	6/25/2008	6/25/2008 03:15 PM	Patient01	Hospital	Image	Test
Appointment	Gerry Anderton	Not Started	(2) Normal	6/26/2008	6/26/2008 09:00 AM	Patient02	Hospital		Test
Appointment	Gerry Anderton	Not Started	(2) Normal	6/27/2008	6/27/2008 01:30 PM	Patient03	Hospital		Test

Table E 4 Surgery appointments WSS list

Title	Assigned to	Status	Priority	Due Date	Appointment time	PatientID	Location	Info About Appointment	Appointment Type

Appoint-ment	Walter Reed	Not Started	(2) Normal	6/26/2008	6/25/2008 10:15 AM	Patient03	Hospital		Proceedure
Appoint-ment	Walter Reed	Not Started	(2) Normal	6/4/2008	6/3/2008 10:00 AM	Patient02	Hospital		Proceedure
Appoint-ment		Not Started	(2) Normal		6/6/2008 08:00 AM	Patient01	Hospital		Referral

Table E 5 Cardiology appointments WSS list

Title	As-signed to	Status	Prior-ity	Due Date	Ap-pointment time	PatientID	Loca-tion	Info About Ap-pointment	Appointment Type
Appoint-ment	Jane Goodhart	Not Started	(2) Normal	5/7/2008	6/16/2008 02:20 PM	Patient01	Hospital		Test
Appoint-ment	Jane Goodhart	Not Started	(2) Normal	6/10/2008	6/9/2008 09:00 AM	Patient02	Hospital		Test
Appoint-ment	Jane Goodhart	Not Started	(2) Normal	5/21/2008	5/17/2008 12:00 AM	Patient03	Hospital		Test

Table E 6 GP appointments WSS list

Title	As-signed to	Status	Prior-ity	Due Date	Ap-pointment time	PatientID	Loca-tion	Info About Ap-pointment	Appointment Type
Appoint-ment		Not Started	(2) Normal	6/15/2008	6/15/2008 12:00 AM	Patient01	Grange Medical		Consultation
Appoint-ment		Not Started	(2) Normal	6/16/2008	6/16/2008 03:00 PM	Patient02	Grange Medical		Consultation
Appoint-ment	Mark Thomson	Not Started	(2) Normal	6/25/2008	6/25/2008 10:00 AM	Patient01	Grange Medical		Consultation

Appendix F Appendix 5

F.1 Alternate use for BDC data HCP-Centred

Patients, in this approach, were not represented as users. Data about the patients was stored on SharePoint lists and the main consumers of the data would be HCPs and not the patients themselves. The same BDC data was also used in this approach, but in different way.

In this method, patients did not have user rights, they did not have access nor have a profile or MY Site, patient attributes data held in AD as shown in Appendix section A.1, was stored in a custom SharePoint list, the custom list data and BDC data obtained in Section 5.4, displayed on an arbitrary WSS 3.0 web part page within the system by HCP's using connecting web parts.

F.2 SharePoint Patient List Creation

A custom list entitled "Patients" was created in the lists container for the Top Level Site (TLS). A site column called "PatientID" was created. Other site columns "DataFirstName", "DataSurname" and "DataEmail" were also created. The site columns were named this way to prevent confusion with the mapped profile attributes of AD. Site columns, matching the additional content types were then created and placed into a custom site column group "CustomCommunity". The custom list was then populated with Patient Information as shown in **Error! Reference source not found.**.

Table F 1 Patient WSS list

Title	PatientID	DataEmail	Data-FirstName	DataSecond-Name
PatientList	Patient01	Patient01 @stirlinghealthcare.com	Janet	Taylor
PatientList	Patient02	Patient02 @stirlinghealthcare.com	Bob	Moore
PatientList	Patient03	Patient03 @stirlinghealthcare.com	Karen	Anderson

Additional default columns "Created by" and "Modified By" used for auditing purposes, were present on the list but not used. The "Title" column is required to link to an individual list item.

F.3 Web Part page Creation

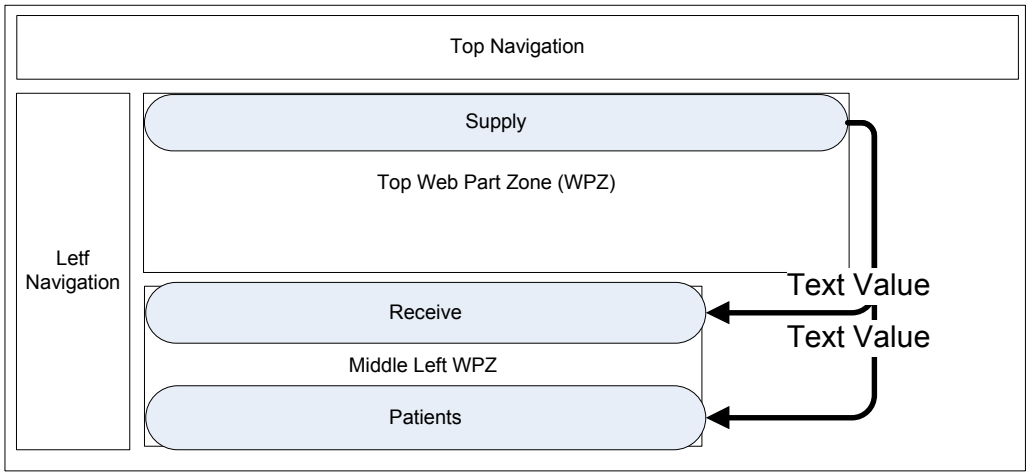
A web part page entitled "Patients" was created in the "pages" document library of the TLS with the URL http://community.stirlinghealthcare.com:81/pages/patientlookup.aspx ;

F.4 Web Part Implementation

The page was then edited; a text filter web part entitled "Supply" was then added to the top Web Part Zone (WPZ) of the newly created page. A BDC item web part entitled "receive" was then added to middle left WPZ of the WSS 3.0 page. When a list is created a web part for the list automatically created. The "Patients" list web part was also added to the middle left WPZ of the page. The security of the page was verified to allow all HCPs contributory (section) rights to the page.

The “receive” web part was configured to look at the scu_portal, entity created in section(section). The Supply web part was then configured to supply the value of the string filter to both the “Receive” web part and the “Patients” list web part(see fig).

Patients Page with Connecting Web Parts



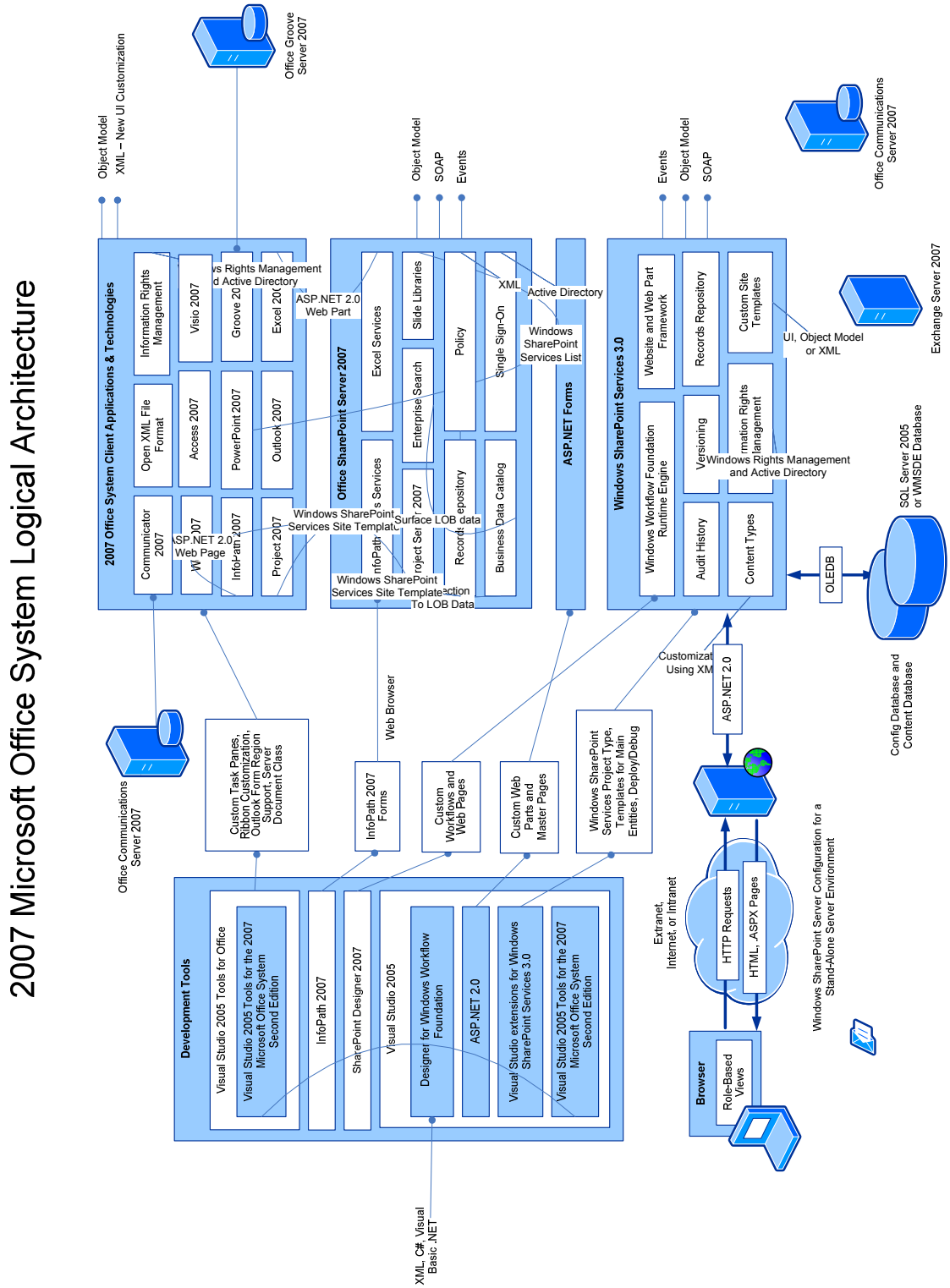
F.5 Testing and Verification

The WSS 3.0 site was then published for all users to see and the system was logged onto using the HCP accounts. The patientlookup.aspx page was visited using the HCP accounts. A filter value of “Patient01” added to the filter box and the go button used. The BDC disparate data was then viewed in the receive web part as shown in fig.

Screen Shot of Results

Appendix G Appendix G

Moss Architecture Visio Drawing (Microsoft, 2005)



Appendix H

H.1 Workflow implementation Class code

```
namespace refrrals2
{
   public sealed partial class Workflow1 : StateMachineWorkflowActivity
   {
      public Workflow1()
      {
         InitializeComponent();
      }


      public SPWorkflowActivationProperties workflowProperties = new SPWorkflowActivationProperties();
      public SPWorkflowTaskProperties createTask1_TaskProperties1 = new Microsoft.SharePoint.Workflow.SPWorkflowTaskProperties();
      public SPWorkflowTaskProperties onTaskChanged1_AfterProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
      public SPWorkflowTaskProperties onTaskChanged1_BeforeProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
      public System.Collections.Specialized.HybridDictionary createTask_SpecialPermissions = new Sys-
tem.Collections.Specialized.HybridDictionary();
      //Action; When the setTask state is reached, the stateInitialiasation activity runs this methods to set the task. The empty method is cre-
ated by dragging a Create Task activity onto the VS2008 design surface
      private void createTask1_MethodInvoking(object sender, EventArgs e)
      {
         SPListItem currentItem = workflowProperties.Item;
         char delim = ('#');
         string s = currentItem["PatientUID"].ToString();
         string t = currentItem["Created By"].ToString();
         string[] created = t.Split(delim);
         string[] words = s.Split(delim);


         createTask1.TaskId = Guid.NewGuid();
         //createTask1.TaskProperties.AssignedTo = @"office\SurgeonSecretary";
         createTask1.TaskProperties.Title = "New appointment referrral";
         createTask1.TaskProperties.Description = "This is an appointment transaction Task";
         createTask1.TaskProperties.DueDate = DateTime.Now.AddDays(2);


         createTask1.TaskProperties.ExtendedProperties["PatientUID"] = @"office\" + words[1];
         createTask1.TaskProperties.ExtendedProperties["Appointment time"] = System.DateTime.Now;
         createTask1.TaskProperties.ExtendedProperties["Appointment Type"] = currentItem["Appointment Type"].ToString();

         createTask1.TaskProperties.HasCustomEmailBody = false;
         createTask1.TaskProperties.EmailBody = "<p>Patient</p> " + words[1] +
             " has been reffred by " + created[1] + " on " +
            System.DateTime.Now.ToString() + ".</p><br><p>Please " + returnTaskUrls("1") + " and add an appointment time and set the
task's status to 'Waiting for HCP' before the due date of " +
            createTask1.TaskProperties.DueDate.ToShortDateString() + "</p><p>Thank you</p>";
         createTask1.TaskProperties.SendEmailNotification = false;
      }


      //Action; Assigns taks to different user
      private void codeActivity1_ExecuteCode(object sender, EventArgs e)
      {
         //Undesirable hardcoding
         createTask1_TaskProperties1.AssignedTo = @"office\Surgeon";
      }
      //Action; Updates the Surgery Referrals workflow item to reflect the workflow progress
      private void codeActivity2_ExecuteCode(object sender, EventArgs e)
      {


         Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
         string status = onTaskChanged1_AfterProperties1.ExtendedProperties[taskListID].ToString();

         SPListItem currentItem = workflowProperties.Item;


         currentItem["Status"] = status;
```

```
        currentItem.Update();

    }
    //Predicate; Checks the decision the HCP made consequent actions are then performed
    private void statusCheck(object sender, ConditionalEventArgs e)
    {

        Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
        string status = onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString();

        if (onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("HCP Accept"))
        {
            e.Result = true;
        }
        else
        {
            e.Result = false;
        }
    }
    //Test point for developer ques
    private void codeActivity3_ExecuteCode(object sender, EventArgs e)
    {
        string breakthere;
    }
    //Updates the workflow item to reflect workflow progress
    private void setAppointmentStatus_ExecuteCode(object sender, EventArgs e)
    {

        Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
        string status = onStatusToHCP_AfterProperties1.ExtendedProperties[taskListID].ToString();


        SPListItem currentItem = workflowProperties.Item;


        currentItem["Status"] = status;
        currentItem.Update();
    }

    public SPWorkflowTaskProperties onStatusToHCP_BeforeProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public SPWorkflowTaskProperties onStatusToHCP_AfterProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public SPWorkflowTaskProperties onHCPAction_AfterProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public SPWorkflowTaskProperties onHCPAction_BeforeProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public String sendHcpMail_Body1 = default(System.String);

    //Action; Updates the workflow item to reflect workflow progress
    private void setListStatusWFA_ExecuteCode(object sender, EventArgs e)
    {
        Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
        string status = onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString();

        Guid assignedToID = workflowProperties.TaskList.Fields["Assigned To"].Id;
        string HCPUID = onHCPAction_AfterProperties1.ExtendedProperties[assignedToID].ToString();
        char delim = ('#');
        string[] words = HCPUID.Split(delim);


        Guid HCPID = workflowProperties.TaskList.Fields["HCPUID"].Id;


        words = null;
        SPListItem currentItem = workflowProperties.Item;


        currentItem["Status"] = status;
        currentItem.Update();
    }

    private void codeActivity2_ExecuteCode_1(object sender, EventArgs e)
    {
```

```
    }


    //Predicate; Checks the decision the patient made consequent actions are then performed
    private void acceptOrRejectApp(object sender, ConditionalEventArgs e)
    {
      Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
      string status = onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString();

      if (onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Patient Accept"))
      {
        e.Result = true;
      }
      else
      {
        e.Result = false;
      }
    }

    public SPWorkflowTaskProperties onPatientChange_AfterProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public SPWorkflowTaskProperties onPatientChange_BeforeProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    // Predicate, asses the status of the workflow to establish what action to take. The event this predicates is used for
    // is a change to the the task when the workflow is in the waitingForAppointment state.
    private void rejectOrCancel(object sender, ConditionalEventArgs e)
    {
      Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
      string status = onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString();

      if (onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Patient Reject"))
      {
        e.Result = true;
      }
      else
      {
        e.Result = false;
      }
    }

    private void reachedWaitingforAPP_ExecuteCode(object sender, EventArgs e)
    {
      //SPListItem currentItem = workflowProperties.Item;
      // createTask1_TaskProperties1.AssignedTo = @currentItem["PatientUID"].ToString();


    }
    //Test
    private void codeActivity2_ExecuteCode_2(object sender, EventArgs e)
    {

    }

    public SPWorkflowTaskProperties updateTask1_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
    public SPWorkflowTaskProperties setAssignedTo_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();

    //Action; Sets the task to a different user changes the due date
    private void setAssignedTo_MethodInvoking(object sender, EventArgs e)
    {
      //udesirable hardcoded user
      setAssignedTo.TaskProperties.AssignedTo = @"office\surgeon";
      setAssignedTo.TaskProperties.ExtendedProperties["HCPUID"] = setAssignedTo.TaskProperties.AssignedTo.ToString();
      setAssignedTo.TaskProperties.DueDate = System.DateTime.Now.AddDays(2);
      setAssignedTo.TaskProperties.Title = "Please Approve Proposed Appointment";

    }
    //Action; updates the task when the waitForAppointment state is reached
    private void updateTask1_MethodInvoking(object sender, EventArgs e)
    {
      Guid stusId = workflowProperties.TaskList.Fields["Status"].Id;


      SPListItem currentItem = workflowProperties.Item;
```

```
            SPList list = currentItem.ParentList;


            char delim = ('#');
            string s = currentItem["PatientUID"].ToString();
            string siteName = list.ParentWeb.Name;
            SPGroupCollection groups = list.ParentWeb.Groups;
            SPGroup visitorsGroup = groups[siteName + " Visitors"];


            string[] words = s.Split(delim);
            try
            {
               visitorsGroup.AddUser("OFFICE\\" + words[1], "", "", "");
            }
            catch (SPException)
            {hnhnghn
}

            //updateTask1_TaskProperties1.ExtendedProperties["Information about your appointment"] = testText;

            updateTask1_TaskProperties1.AssignedTo = @"office\"+words[1]; //+currentItem["PatientUID"].ToString();
            //updateTask1_TaskProperties1. = @"office\"+words[1]
            words = null;

         }
         //Predicate; determine if an HCP has rejected a appointment
         private void checkReject(object sender, ConditionalEventArgs e)
         {
            Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
            string status = onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString();

            if (onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("HCP Reject"))
            {
               e.Result = true;
            }
            else
            {
               e.Result = false;
            }
         }
         //Predicate; error determination
         private void checkErrorsEvents(object sender, ConditionalEventArgs e)
         {
            Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
            string status = onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString();

            if (onHCPAction_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("HCP Reject") || onHCPAc-
tion_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("HCP Accept") || onHCPAc-
tion_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Cancelled"))
            {
               e.Result = true;
            }
            else
            {
               e.Result = false;
            }
         }
        //Action; sets workflow item to reflect workflow progress
         private void setListStatusWFD_ExecuteCode(object sender, EventArgs e)
         {
            SPListItem currentItem = workflowProperties.Item;
            currentItem["Status"] = "Waiting For HCP";
            currentItem.Update();
         }

         //Predicate; checks the workflow progress
         private void checkPatientActionStat(object sender, ConditionalEventArgs e)
         {
            Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
            string status = onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString();
```

```
            if (onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Patient Reject") || onPa-
tientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Patient Accept") || onPa-
tientChange_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Cancelled"))
            {
               e.Result = true;
            }
            else
            {
               e.Result = false;
            }
         }
        //Action; update list to rreflect workflow
         private void setListStatusPatientRej_ExecuteCode(object sender, EventArgs e)
         {
            SPListItem currentItem = workflowProperties.Item;


            currentItem["Status"] = "Waiting on Dep";
            currentItem.Update();
         }
         //Action; update list to rreflect workflow
         private void setListsItem_ExecuteCode(object sender, EventArgs e)
         {
            SPListItem currentItem = workflowProperties.Item;


            currentItem["Status"] = "Waiting on Dep";
            currentItem.Update();
         }
         //Action; update list to rreflect workflow
         private void setErrorPatientAction_ExecuteCode(object sender, EventArgs e)
         {
            SPListItem currentItem = workflowProperties.Item;


            currentItem["Status"] = "HCP Accept";
            currentItem.Update();
         }
         //Action; update task list to rreflect workflow
         private void onGetStateUpdate_MethodInvoking(object sender, EventArgs e)
         {
           //undesirable hardcoding
           onGetStateUpdate_TaskProperties1.AssignedTo = @"office\SurgeonSecretary";
           onGetStateUpdate_TaskProperties1.DueDate = System.DateTime.Now.AddDays(2);
         }
        //test
         private void updateTaskActivity_ExecuteCode(object sender, EventArgs e)
         {


            //createTask1.TaskProperties.AssignedTo = @"office\SurgeonSecretary";
         }

         public SPWorkflowTaskProperties onGetStateUpdate_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
         public Guid createTask1_TaskId1 = default(System.Guid);

         //Predicate; check to see if appointment has been cancelled
         private void depCancelled(object sender, ConditionalEventArgs e)
         {
            Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
            string status = onStatusToHCP_AfterProperties1.ExtendedProperties[taskListID].ToString();

            if (onStatusToHCP_AfterProperties1.ExtendedProperties[taskListID].ToString().Equals("Cancelled"))
            {
               e.Result = true;
            }
            else
            {
               e.Result = false;
            }
         }
         //Action; update task list to rreflect workflow
         private void setListCancelled_ExecuteCode(object sender, EventArgs e)
         {
```

```
    SPListItem currentItem = workflowProperties.Item;


    currentItem["Status"] = "Cancelled";
    currentItem.Update();
  }
  //Action; update task list to rreflect workflow
  private void setListCycle(object sender, EventArgs e)
  {
    SPListItem currentItem = workflowProperties.Item;


    currentItem["Status"] = "Waiting For HCP";
    currentItem.Update();
  }

  //Predicate; check to see appointment is complete
  private void isAppointmentComplete(object sender, ConditionalEventArgs e)
  {
    Guid statusID = workflowProperties.TaskList.Fields["Status"].Id;
    string statusComplete = onPatientChange_AfterProperties1.ExtendedProperties[statusID].ToString();

    if(statusComplete.Equals("Complete"))
    {
      e.Result = true;
    }
    else
    {
      e.Result = false;
    }

  }
 //test
  private void setAppListItem(object sender, EventArgs e)
  {


  }
  //Actions; Completes the workflow, add history items to department and patient history list and remove users
  private void CompleteTaskItem(object sender, EventArgs e)
  {
    Guid hcpID = workflowProperties.TaskList.Fields["HCPUID"].Id;
    Guid locGuid = workflowProperties.TaskList.Fields["Location"].Id;
    string location = onPatientChange_AfterProperties1.ExtendedProperties[locGuid].ToString();
    string hcpUID = onPatientChange_AfterProperties1.ExtendedProperties[hcpID].ToString();
    Guid appTimeID = workflowProperties.TaskList.Fields["Appointment time"].Id;
    string appTime = onPatientChange_AfterProperties1.ExtendedProperties[appTimeID].ToString();
    Guid sttusId = workflowProperties.TaskList.Fields["Status"].Id;
    string finalStatus = onPatientChange_AfterProperties1.ExtendedProperties[sttusId].ToString();

    SPListItem currentItem = workflowProperties.Item;
    char delim = ('#');
    string s = currentItem["PatientUID"].ToString();
    string[] col = s.Split(delim);
    string pat = col[1];
    string[] hcpCol = hcpUID.Split(delim);


    SPSite Site = new SPSite("http://community.stirlinghealthcare.com:81/personal/" + pat);
    SPWeb pWeb = Site.OpenWeb();




    SPSite parSite = new SPSite("http://community.stirlinghealthcare.com:81/surgery");
    SPWeb web = parSite.OpenWeb();
    SPUser user = web.AllUsers[@"" + hcpUID];
    string userName = user.ID + ";#" + user.Name;



    SPListTemplateCollection templates = parSite.GetCustomListTemplates(web);
    SPListCollection pWebLists = pWeb.Lists;
    SPListTemplate template = templates["appHistory"];
    //SPUser patient = web.AllUsers["OFFICE\\" + col[1]];
```

```
            try
            {

               pWeb.Lists.Add("Appointment History", "Your Appointment History", template);
               pWeb.Update();
            }
            catch (SPException)
            {
            }


            SPList appHistSurg = web.Lists["Appointment History"];

            SPListItem surgListItem = appHistSurg.Items.Add();

            SPList appList = pWebLists["Appointment History"];

            SPListItem pListItem = appList.Items.Add();

            pListItem["Title"] = "Past Appointment";
            surgListItem["Title"] = "Past Appointment";
            pListItem["Status"] = finalStatus;
            surgListItem["Status"] = finalStatus;
            pListItem["Patient Name With ID"] = s;
            surgListItem["Patient Name With ID"] = s;
            pListItem["HCP Name with ID"] = userName;
            surgListItem["HCP Name with ID"] = userName;
            pListItem["Location"] = location; //currentItem["Location"].ToString();
            surgListItem["Location"] = location;// currentItem["Location"].ToString();
            pListItem["Appointment time"] = System.DateTime.Parse(appTime);
            surgListItem["Appointment time"] = System.DateTime.Parse(appTime);
            pListItem.Update();
            surgListItem.Update();

            Site.Close();
            parSite.Close();
            web.Dispose();
            pWeb.Dispose();

        }

        public SPWorkflowTaskProperties updateTask2_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();

        private void updateTask2_MethodInvoking(object sender, EventArgs e)
        {

        }

        public SPWorkflowTaskProperties onPatientAccept_BeforeProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
        public SPWorkflowTaskProperties onPatientAccept_AfterProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();

        private void updatePatientAccept_MethodInvoking(object sender, EventArgs e)
        {
            Guid taskListID = workflowProperties.TaskList.Fields["Status"].Id;
            string status = onPatientChange_AfterProperties1.ExtendedProperties[taskListID].ToString();
            Guid appTimeID = workflowProperties.TaskList.Fields["Appointment time"].Id;
            string appTime = onPatientChange_AfterProperties1.ExtendedProperties[appTimeID].ToString();
            System.DateTime date = System.DateTime.Parse(appTime);
               updatePatientAccept_TaskProperties1.Title = "Appointment";
               updatePatientAccept_TaskProperties1.DueDate = date;
            SPListItem Item = workflowProperties.Item;
            Item["Status"] = status;
            Item.Update();
        }

        public SPWorkflowTaskProperties updatePatientAccept_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
        public SPWorkflowTaskProperties updateCancelled_TaskProperties1 = new Micro-
soft.SharePoint.Workflow.SPWorkflowTaskProperties();
```

```
        private void setSendInitialMail(object sender, EventArgs e)
        {
           /*Guid taskItemID = workflowProperties.TaskList.Fields["ID"].Id;

           SPListItem currentItem = workflowProperties.Item;
           sendInitialEmail.To = @"OFFICE\SurgeonSecretary";
           sendInitialEmail.Subject = "New Appointment Referral";
           sendInitialEmail.Body = "<p>Patient</p> "; /* +currentItem["PatientUID"].ToString() +
              " has been reffred by " + currentItem["Created By"].ToString() + " on " +
              System.DateTime.Now.ToString() + ".</p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) + " and add an
appointment time before the due date of " +
              createTask1.TaskProperties.DueDate.ToShortDateString() + "</p><p>Thank you</p>";*/
        }

       /*   private void completeTask1_MethodInvoking(object sender, EventArgs e)
          {
             Guid hcpID = workflowProperties.TaskList.Fields["HCPUID"].Id;
             string hcpUID = onPatientChange_AfterProperties1.ExtendedProperties[hcpID].ToString();
             //   onTaskProperties.ExtendedProperties[hcpID].ToString();

             SPListItem currentItem = workflowProperties.Item;
             char delim = ('#');
             string s = currentItem["PatientUID"].ToString();
             string[] col = s.Split(delim);
             string pat = col[1];
             string[] hcpCol = hcpUID.Split(delim);


             SPSite Site = new SPSite("http://community.stirlinghealthcare.com:81/personal/" + pat);
             SPWeb pWeb = Site.OpenWeb();




             SPSite parSite = new SPSite("http://community.stirlinghealthcare.com:81/surgery");
             SPWeb web = parSite.OpenWeb();
             SPListTemplateCollection templates = parSite.GetCustomListTemplates(web);
             SPListCollection pWebLists = pWeb.Lists;
             SPListTemplate template = templates["geniricAppointments"];
             //SPUser patient = web.AllUsers["OFFICE\\" + col[1]];


             try
             {

                pWeb.Lists.Add("Appointment History", "Your Appointment History", template);
                pWeb.Update();
             }
             catch (SPException)
             {
             }



             SPList appList = pWebLists["Appointment History"];
             SPListItem pListItem = appList.Items.Add();
             pListItem["Title"] = "Past Appointment";
             pListItem["PatientUID"] = hcpUID;
             pListItem["HCPUID"] = hcpUID;
             pListItem.Update();
             pListItem["Appointment time"] = createTask1_TaskProperties1.ExtendedProperties["Appointment time"].ToString();
             pListItem["Location"] = createTask1_TaskProperties1.ExtendedProperties["Location"].ToString();

          }
    */
        private string returnTaskUrls(string taskId)
          {
             //Use this to return the relevant URL's for use in e-mail of tasks.
             string weburl = workflowProperties.WebUrl;
             string taskUrl = "/_layouts/WrkTaskIP.aspx?List=";
             string listId = workflowProperties.TaskListId.ToString();
             string fulltaskUrl = weburl + taskUrl + listId + "&ID=" + taskId;
             string htmlTaskLink = "<a href=" + "\"" + fulltaskUrl + "\"" + ">Click here to open the task directly in SharePoint</A>";

             return htmlTaskLink;
```

```
        }


    private void setSendHCPMail(object sender, EventArgs e)
    {

        //sendHcpMail.To = @"OFFICE\" + created[1]; //onStatusToHCP_AfterProperties1.AssignedTo;//
        //sendHcpMail.Subject = onStatusToHCP_AfterProperties1.Title;
       /* sendHcpMail.Body = ( "<p>Patient</p> " + words[1] +
            " has been reffred by " + created[1] + " on " +
           System.DateTime.Now.ToString() + ".</p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) +
           ". If the appointment time suitable plasea set the task's status to 'HCP Accept' or if the appointment time of " + Sys-
tem.DateTime.Parse(t) +
           " is not suitable, set the task's status to 'HCP Reject' and new arragement will be made.</p><p>Please could you complete this
action before the due date of " +
           onStatusToHCP_AfterProperties1.DueDate.ToShortDateString() + "</p><p>Thank you</p>");
        */
    //   sendEmail1_Body1 = "Hello world";
     //    sendEmail1_Subject1 = "Title";
      //  sendEmail1_To1 = @"surgeon@sitrlinghealthcare.com";
    }
    public String sendEmail1_Body1 = default(System.String);
    public String sendEmail1_To1 = default(System.String);
    public String sendEmail1_Subject1 = default(System.String);
    public Int32 createTask1_ListItemId1 = default(System.Int32);

    //Action; send email to the personed assigned to the task
    private void sendEmail1_MethodInvoking(object sender, EventArgs e)
    {

        Guid appointTimeID = workflowProperties.TaskList.Fields["Appointment time"].Id;
        string t = onStatusToHCP_AfterProperties1.ExtendedProperties[appointTimeID].ToString();
        Guid assID = workflowProperties.TaskList.Fields["Assigned To"].Id;
        string to = setAssignedTo_TaskProperties1.ExtendedProperties[assID].ToString();
        SPListItem currentItem = workflowProperties.Item;
        char delim = ('#');
        char splitSlash = ('\\');
        string s = currentItem["PatientUID"].ToString();
        string u = currentItem["Created By"].ToString();
        string[] ass = t.Split(delim);
        string[] created = u.Split(delim);
        string[] words = s.Split(delim);
        string[] assignedToArray = to.Split(splitSlash);
        this.sendEmail1_Body1 =  "<p>Dear "+ to+"</p> " + words[1] +
            " has been reffred by " + created[1] + " on " +
           System.DateTime.Now.ToString() + ". </p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) +
           ". If the appointment time suitable plasea set the task's status to 'HCP Accept' or if the appointment time of " + Sys-
tem.DateTime.Parse(t) +
           " is not suitable, set the task's status to 'HCP Reject' and new arragement will be made.</p><p>Please could you complete this
action before the due date of " +
           onStatusToHCP_AfterProperties1.DueDate.ToShortDateString() + "</p><p>Thank you</p>";
        this.sendEmail1_Subject1 = setAssignedTo_TaskProperties1.Title;
        this.sendEmail1_To1 = assignedToArray[1] + "@stirlinghealthcare.com";



    }

    //Action; This email activity sends the email to the SurgeonSecretary user alerting them of a new task.
    private void sendInitialEmail_MethodInvoking(object sender, EventArgs e)
    {


        SPListItem currentItem = workflowProperties.Item;
        char delim = ('#');

        string s = currentItem["PatientUID"].ToString();
        string u = currentItem["Created By"].ToString();
        //string[] ass = t.Split(delim);
        string[] created = u.Split(delim);
        string[] words = s.Split(delim);

        this.sendEmail1_Body1 = "<p>" + words[1] +
            " has been reffred by " + created[1] + " on " +
```

```
            System.DateTime.Now.ToString() + ".</p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) + " and add an
appointment time before the due date of " +
            System.DateTime.Now.AddDays(2) + "</p><p>Thank you</p>";
          this.sendEmail1_Subject1 = "New Appointment Referral";
          this.sendEmail1_To1 = "surgeonsecretary@stirlinghealthcare.com";
       }

       //Action; send email to patient after HCP has accepted
       private void sendAcceptenceToPatientEmail_MethodInvoking(object sender, EventArgs e)
       {
          Guid appointTimeID = workflowProperties.TaskList.Fields["Appointment time"].Id;
          string t = onStatusToHCP_AfterProperties1.ExtendedProperties[appointTimeID].ToString();
          Guid assID = workflowProperties.TaskList.Fields["Assigned To"].Id;
          string to = setAssignedTo_TaskProperties1.ExtendedProperties[assID].ToString();
          SPListItem currentItem = workflowProperties.Item;
          char delim = ('#');
          char splitSlash = (\\');
          string s = currentItem["PatientUID"].ToString();
          string u = currentItem["Created By"].ToString();
          string[] ass = t.Split(delim);
          string[] created = u.Split(delim);
          string[] words = s.Split(delim);
          string[] assignedToArray = to.Split(splitSlash);
          this.sendEmail1_Body1 = "<p>Dear " + to +
            "</p> An appointment has been made for the time of"  + t + " on " +
            System.DateTime.Now.ToString() + ". </p><br><p>Please " + returnTaskUrls(createTask1.ListItemId.ToString()) +
            ". If the appointment time suitable please set the task's status to 'Patient Accept' or if the appointment time of " + Sys-
tem.DateTime.Parse(t) +
            " is not suitable, set the task's status to 'Patient Reject' and new arragements will be made.</p><p>Please could you complete this
action before the due date of " +
            onHCPAction_AfterProperties1.DueDate.ToShortDateString() + "</p><p>Thank you</p>";
          this.sendEmail1_Subject1 = setAssignedTo_TaskProperties1.Title;
          this.sendEmail1_To1 = words[1] + "@stirlinghealthcare.com";
       }

       private void completeTask1_MethodInvoking(object sender, EventArgs e)
       {

       }

       public static DependencyProperty createTask1_SpecialPermissions1Property = DependencyProp-
erty.Register("createTask1_SpecialPermissions1", typeof(System.Collections.Specialized.HybridDictionary), typeof(refrrals2.Workflow1));

       //This code was an attempt implement hardened workflow security by only allowing access to task list items, to the person the task is
assigned.
       //See dynamic security model.
       [DesignerSerializationVisibilityAttribute(DesignerSerializationVisibility.Visible)]
       [BrowsableAttribute(true)]
       [CategoryAttribute("Misc")]
       public System.Collections.Specialized.HybridDictionary createTask1_SpecialPermissions1
       {
          get
          {
             return ((Sys-
tem.Collections.Specialized.HybridDictionary)(base.GetValue(refrrals2.Workflow1.createTask1_SpecialPermissions1Property)));
          }
          set
          {
             base.SetValue(refrrals2.Workflow1.createTask1_SpecialPermissions1Property, value);
          }
       }









    }
}
```

Appendix I

I.1 Interview Transcripts

The following text entries are transcripts of interviews which took place on the 1st September and 4th September 2008. The interviews were conducted first with Dr. Christoph Thuemmler (CT), Consultant Physician, General Practitioner and IT Researcher for Healthcare. The second interview is with Professor William Buchanan (BB).

I.2 Interview 1 Christpoh Thuemmler

MATT: OK, this is an interview with Christoph Thuemmler about my project which is a patient administration integration and work flow system. First of all, I need to ask you your name and the role in which you are associated with, a) the university and b) the NHS.

CT: My name is Dr Christoph Thuemmler and I am a consultant physician and I am a GP. I work for the Western Isles Health Board and I have been at Napier University for 6 months.

MATT: Now, you said that your role is to bring IT and healthcare together. Can you tell me some of the projects in relation to this that you have been working on recently?

CT: Yes. I mean, I've been working for the NHS Out of Hours initiative and I've been working on the patient flow modelling, and I'm working on the Scottish early warning scores. and applying logic algorythms to these kinds of structures.

MATT: As a general context statement, do you believe that there is an issue here with IT and healthcare and, if you do, could you describe some of the causes of these issues?

CT: Yes, I think there is an issue in healthcare. We know from investigations back in 2005 which were published in the British Medical Journal that in general terms the NHS IT infrastructures are suffering from severe underfunding, so there is definitely a problem with that, and there is definitely a problem with linking different systems and a compatibility of different systems in IT, and I think that these systems are very hard to individualise, to customise, so there are definitely issues about how to really get the system fit for the individual setting and the individual patient.

MATT: The national program for IT, originally the IMT strategy of 1992, indicated some of the issues you just mentioned such as integration and I was wondering how do you think those strategies have met their aims, in particular 1992 and 1998 IMT and the new national program for IT?

CT: I think if you really apply hard criteria, then these initiatives failed or might have good intentions but due to the underfunding they are basically lagging behind developments. We are in a situation now where the last Trusts are getting their wireless networks when, in fact, they should have specialist systems already implemented. So I think there are huge deficits and I think it is key that the development goes further.

MATT: The national program for IT stipulates that systems should be patient-centric. What is your interpretation of the term patient-centric.

CT: I think patient-centric means to me that the patient is part of an intricately integrated network and ideally this should be grouped around the patient and should be a network that is not only accessible to the professionals, consultants, nurses who are dealing with the patient at the moment, but I think it should be accessible to all of the team members who are taking part in the care of the patient, and it should be a scalable system so the patient should in the centre of the effort and the network should be spun around the patient and should be accessible from every point, and information should be exchangeable and should be accessible from everywhere.

MATT: Brilliant, that's excellent, thanks. In you past dealings with the NHS or your involvement with the NHS, have you used any of these "patient-centric" systems and, if you have, how would you assess their usefulness?

CT: I haven't worked in environments that were really built up like that. I think the idea of patient-centred care here in the UK is relatively new although existing for over 10-15 years; but there are actually no systems implemented that work like that.

MATT: None of the systems are truly patient-centric.

CT: Yes, that's correct.

MATT: OK, I'm going to pause now to demo the system with you and then we'll come back after with some questions about it.

Christoph, you've just had a look at the system and I'm going to ask you a few questions about the system and I would like you to expand on these questions if you can. You've seen the approaches to the design and the use of the system, and particularly, you've seen the integration and data for disparate sources. What real-world data types would you consider to be useful for integration?

CT: Well I think it is important to have access not only to the medical information but, given the demographic developments in society with more and more old people around who have different requirements than just medical input, we need to have access to social work, for example social work data, care home information – all these kinds of things should be considered and we should be able to collect these inputs. That means having access to the GP,

having access to community nurses, having access to pharmacy dispensary, social worker, home care, etc., etc. And on the other hand, I think there could be at some stage input from sensors from devices placed in intelligent homes and all these kinds of things.

MATT: Thank you. You've seen the integration page or the consolidation or federation page on the system whereby the patient could view the current state of their case and they can also go and see doctors and look at skills etc. Do you think this increases the visibility of the NHS for patients and, if so, how?

CT: Yes, definitely. I think it is very important that patients get a relatively clear view on what's happening. At the moment, it is nearly impossible for patients to oversee in real time what is actually going on with their care and where they stand. I think it is much better if visibility is better. It enables the patient by the end of the day to add and contribute to his own management, to improve the management of his problems which I think makes work easier for all parties involved.

MATT: Do you think the system of logging work flow activity automatically can reduce administrative overheads on healthcare staff?

CT: I think it can because lots of the administrative duties are with the nurses, not with the administrators, and of course it takes away the time that nurses have to spend on other patients. So I think the system will be very helpful, not only for nurses but I think for doctors as well and we can avoid duplicity in all these kinds of things, so I think it is a good thing.

MATT: Overall, do you think the system which was produced is patient-centric?

CT: I think it is. The patient is definitely strategically placed closer to the centre than we see in the current systems. I think for the time being we can be happy with the position of the patient in the system.

MATT: Can you envisage a use for this system or the approaches used in the design of this system for further research?

CT: Definitely. I think that we might see systems coming up which are able to link with all kinds of sources of information in real time, especially in connection with the IPV6 which will allow real time computing. So that means if the system would be able to link to routers and would be able to link with the periphery wire links and lines that enable real time computing, and if you could connect it to remote databases that can actually process the information, we could use this kind of software as a middle way to control this whole real time computing system which would allow real time information.

MATT: What social barriers do you think exist that could inhibit the adoption of such a system?

CT: I think there is a lot of anxiety about security issues which needs to be resolved . . .

MATT: Security?

CT: Yes, data protection in the first place. So, people might be concerned about their personal data. People might be concerned that the system could be accessed by other people with malicious intent. I think there might be resistance within the administration that needs to be overcome. I think it is important that the software will not increase the burden on the administration of the IT administration in healthcare. On the other hand, you know, we will see whether it will increase or will not increase the burden, but I think it is already …….. if there is anxiety around that this *might* increase the burden of the IT administration. So I think these are the major problems.

MATT: Do you think these problems have been inherent in past implementations?

CT: Yes. We know that there are lots of papers around by the Dept of Health or by the NHS, so there are problems with change within the NHS. There are anxieties and the NHS is a notoriously difficult structure if it comes to the implementation of change, but I think this can be overcome.

MATT: That's it. Thank you very much for your participation.

I.3 Interview 2 William (Bill) Buchanan (BB)

MATT: OK, I'm here with Prof Bill Buchanan and he has agreed to evaluate my patient-centric patient administration system which incorporates workflow. First of all, I need to ask you what your name and role are.

BB: My name is Bill Buchanan. I'm Professor of the School of Computing and I lead the Centre for Distributed Computing and Security. We are very much a forward facing research group that tries to tackle major issues.

MATT: Can I ask you, in terms of healthcare, what projects you have been working on recently?

BB: We've been mainly working with Christoph Thuemmler who is a medical consultant. Initially we had a project in Birmingham relating to a piece of identification using RIFD tags We've now enhanced that work into two major themes of using RFID for infection tracking and trace-back and also we're interested in patient flow analysis and possibly in the organisation of patient care, looking into things like bed management and resource allocation.

MATT: In your experience in working with these systems, do you believe that there is an issue with IT and healthcare in general and, if you do, what do you think are the causes of these issues?

BB: We think there's a massive issue with IT and healthcare and we really do believe that IT infrastructures can actually help patients and practitioners but the adoption of it has been fairly poor for many reasons. We think that IT can help the practitioners focus on better diagnosis, better monitoring, and our key focus is trying to collect data that we can use to make things better, to analyse our inefficiencies and try to improve; in fact, I think a major issue is related to the fact that the NHS tends to be fairly regionalised in its structure. IT systems tend to be localised with a massive central store and obviously a key issue is to bring the data together and to make sure that it is useful but also that it is secure, and we have great worries from a security point of view that simple security breaches could cause a lack of understanding of adoption within it, so I think security probably is the number one issue in any great roll-out. We feel that the NHS will tend to centralise data to keep it secure, fundamentally because they think if it is physically kept in the one place then everything is fine and records can take two or three minutes to access centralised structures; we think there are better structures for this; they are un-centralised.

MATT: The national program for IT when it was launched as a different name, it was called the NHS National Plan in the year 2000. It was the first plan that was introduced to the nation as a patient-centric systems. What's your interpretation of the term patient-centric?

BB: Patient-centric – as far as we think, and it's certainly an observation from working within healthcare systems to be able to view them from a practitioner point of view and also from a user point of view as observed in terms of visiting patients. Too much of what goes on is centred around procedures and practitioners themselves, especially at consultant level. Consultants like the idea of things revolving around them and that patients revolve around these consultants and it gives them a position of power which they've actually used greatly. There is very little auditing as far as I can tell, in a procedure if something goes wrong. In most of our jobs these days there is some audit trail to actually show what went wrong and who did it wrong and why it went wrong, and the key thing is to learn from our mistakes and make sure things are done better. We don't necessarily want to go for a league table type approach which tends to be the way the Government goes, but what I am trying to do is identify weaknesses in the processing system. So, too often, practitioners are sitting the especially at consultant level and defining their own procedures and practices and almost become like semi-autonomous self-employed entities .

MATT: Like disparate entities?

BB: . . . disparate entities – they are not integrating with other parts of the system. So from what I can observe, the consultants are making judgements at the minute without looking back at what happened previously. To me . . .

MATT: Is that because the information is generally not available to them or is it just because of the attitude of their work ethic?

BB: For many reasons. It is an old fashioned work ethic and the healthcare system isn't keen to move away from something that works already. They tend to be extremely busy and aren't willing to put in the time and effort to get feedback on these systems to make them better. I think the IT industry itself is to blame to some extent. They tend to go away into their own cubby-holes, develop their systems and then roll them out and they look fine but they haven't involved the nurses and the doctors at the various levels.

MATT: Would you say that's reflected in the current systems available?

BB: The current systems tend to be based on mainframe type architectures. They tend to be terminal driven and tend not to be good from a security point of view simply because of the need for passwords . . .

MATT: Would you say that user names and passwords are decentralised as opposed to intrinsic across the organisation, so, for instance, you've got a system that's developed, does it carry its own security make-up and then people have to log in to each of these systems to get something done?

BB: Yes, there needs to be a joined-up approach to the whole infrastructure. Start with the top level and work down and make sure the security is thought at every single stage. What tends to happen is that there tend to be bridges and the security is defined between the bridge and unfortunately the entity at either side of the bridge can be insecure and data can flow, and often security can get in the way of usability and it shouldn't actually be like that. There is very little analysis done on what data is critical at any point. The timeliness of the data is important so the whole thing isn't thought and I think they take the Lego blocks and try to build the system. It is a much better approach just to start from scratch. Look at the organisation and set it up rather than trying to model the data around that, especially how the patient reacts with the healthcare environment, and the big problem is between primary and secondary tertiary care although the boundaries between that are obviously impossible to understand.

MATT: OK, thanks very much. I'm going to pause now to show you the system that I've created for my dissertation and we'll come back with a few more questions about that.

MATT: you've seen the integration and data for disparate sources. What real-world data types would you consider to be useful for integration?

BB: There are many possible sources of disparate data that could be used in this type of application. In a health care environment it is well known that GPs and Consultants need to make their judgments on a wide range of factors, which are not necessarily related to physiological tests or a visual check. Thus other data could thus related to associated factors such as

their social background and their social integration, where a decision is made on whether to keep an elderly person in hospital might relate to whether a next-of-kin is living nearby, and whether they could be contacted. Also it cannot be always assumed that the records on a person will be complete, thus records for things like medications taken, and their history of usage, would be useful. A confirmation of this from pharmacy records would help validate whether medicines had actually been picked-up. Along with this a personal record might show whether the medicine had actually been taken. The health care system is also dealing with many more cases of confused and disoriented patients, thus access to information on contact points would be useful, in helping to optimize the care of these types of patients. Often, at the present time, this follow-up happens after many days for care, where this system could provide this data quickly so that a patient can be assess when they are admitted into the secondary health care environment.

Another useful source of data might relate to helpful information related to any illnesses that someone might have, which could be "pushed" to the patient, in order that they are better informed. I also think there should be a source of "emergency" data, which relates to integrating data that would not normally be viewed by the HCP, but which could be viewed in an emergency. Thus could relate to a sensitive issue, that the patient did not want anyone to know about, but which is important in an emergency. An example of this might relate to a private operation that was conducted outwith the NHS system, but of which the details could be logged, along with contacts.

MATT: Do you think the system of logging work flow activity automatically can reduce administrative overheads on healthcare staff?

BB:I think workflow is an excellent feature, and one of the most useful in the system. The key factor is that problems in processes can be picked-up at an early stage, such as if a HCP is cancelling a large amount of activities, without giving a sensible reason. The consent based approach is very good as it allows the patient to make decisions on their own circumstances, and they will thus feel more in control. The workflow also simplifies the process, and allows everyone to be kept informed of the status of a given event.

MATT: Do you think this system increases the visibility of the NHS for patients and, if so, how?

The My Site is a good focal point for the patient, and it allows the patient to go to a single place and correspond and keep track of their health care environment. I think the integration with email is also an excellent. Obviously one of the barriers will be the IT-awareness of many users, and the possibility that HCP will feel that they are being monitored for in the operations. This is something, though, which possibly needs to change, as the health care has increasing demands, and with increasing money going into it, but patients still do not often feel part of the system. A survey out today stated that "productivity" had actually fallen in the NHS, which is a stark statistic in the light of increasing funding. Patients using this system will be able to guarantee at least a certain quality of service, such as related to the

time that a HCP takes to respond to an event. Along with this the system cuts out the bureaucracy involved in the current system, where delays of weeks can happen with correspondence, along with obviously problems of patients not reading letters, and so on. With this system the response to events can be traced, and a non-action, would result in further correspondence.

MATT: How would you enhance the system?

BB: Some enhancements might be to setup warning messages, when a patient or HCP did not respond within a given time limit. Perhaps it could relate to an email/SMS message. Create an analysis program which monitored data related to the service level for the HCP and patient, such as average time to respond to an event. Possibly reorganise the structure of the departments to the user, so that they are presented in a more structured way. There maybe needs to be some form of information bank in which users get more information on events, and the medical practice involved. Perhaps this might involve some filtering of the event to make sure that it was relevant to the event. The integration with email/SMS is an important and needs to be carefully managed, especially from a security/cost point-of-view. Provide a bridge to a deeper level of information, which would required privilege levels of access, that could only be used in an emergency situation. Enhance the authentication method, such as integrating with biometric and smart card technologies. Provide enhanced and focused search facilities. Analyse roles and interactions in the NHS, and use this to optimize the workflows, and the roles (especially to enhance security). Understand how HCP perceive such as system, and how it can be optimized for their usage.

OK, overall in your evaluation of the system, do you think it is patient-centric and why?

BB: I think it is definitely patient-centric. It's centric around whoever logs in and they see the data that's relevant to them, so even the healthcare practitioner actually feels that data is revolving around them and should make things easier. All the data is there and the MY Site is excellent because it gives a one stop portal for all the data.

MATT: OK, thanks. Can you envisage a use for this system or the approaches used in the design of this system for further research?

BB: Yes, I think this is exactly the type of system we are trying to create. The biggest problem found in healthcare is trying to link to different disparate databases for certain types of research work. This seems to be the way that integrates into a common framework and we can define the security, we can define bridges, we can link to certain databases without being tied into a specific vendor or manufacturer of certain equipment because the .NET platform is so well used it should be fairly easily to implement this in a healthcare environment.

MATT: OK, thanks very much for your participation.

Appendix J

J.1 Virtual Technology

To allow this research to be passed on in its entirety, virtual technology is used. Virtual technology is software allowing operating systems such as Windows Server 2003 and Lunix Red Hat, to hosted on a single physical machine, the virtualisation layer ports all necessary Input Output busses to emulate hardware transparently. In this context virtual technology allows the entire project to run from a single machine. In addition, subject to licensing arrangements the project can handed over as a set of files and run, locally, in a different location or copied so different teams can work on the same infrastructure. VMWare workstation is used for this project; however, there are other virtual machine technologies available. Figure J 1 shows virtual technology architecture in the context if the infrastructure for this project.

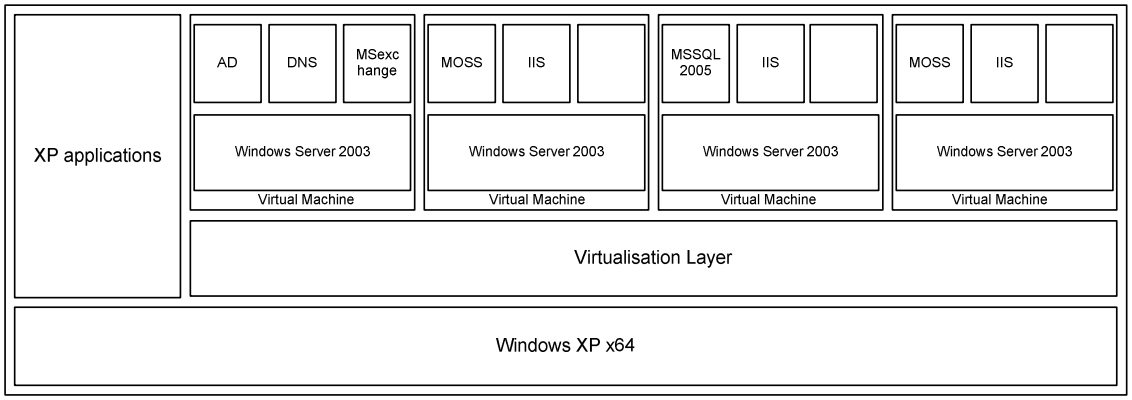


Figure J 1 Virtual technology architecture showing virtual machines used