

Requirements Capture in E-Science Systems

PhD research Proposal

Kate Ho

Social Informatics Cluster,
University of Edinburgh
1 Buccleuch Place,
Edinburgh, EH8 9LW

Abstract

E-science is hailed as a new way of doing science. Its aim is to create an infrastructure which enables collaborative and distributed work, coupled with large computational processing power and data storage facilities. E-Science brings together a number of challenges both social and technical, and most of the time, a combination of both. These include collaborative working, information governance, database linking, security etc. Whilst there is an understanding of some of the challenges that face e-science projects, there has been little attention on the requirements capture process of e-science projects. This thesis aims to examine the requirements specification of the projects and aims to draw out any differences from traditional requirements specification. This research proposal will outline the background to the literature, and how the research will be carried out, progress from the first year of study and proposed work for the next remainder period of study.

Contents

Chapter 1: Introduction	4
1.1. Background – What are e-science systems?	4
1.2. Requirements capture in e-science systems	5
1.3. Research Questions	7
1.4. The Approach – Empirical (Observational) research of System Developers	8
1.5. Organisation of the Proposal	9
Chapter 2: Requirement Capture in E-Science Systems	11
2.1. Introduction	11
2.2. E-Science.....	11
2.2.1. E-Science – The Vision.....	11
2.2.2. What challenges does it face?	13
2.2.3. Characteristics of E-Science Systems	14
2.3. Requirements Capture	17
2.4. Summary	22
Chapter 3: Research Methodology	24
3.1. Introduction	24
3.2. The need for a Qualitative Approach	24
3.3. Selecting Case Studies	26
3.4. Research Case Studies.....	28
3.4.1. Case Study 1	28
3.4.2. Other case studies.....	33
3.5. Understanding the Real World: Data Collection Methods.....	33
3.5.1. Observations Methods/Ethnography	34
3.5.2. Informal and Formal Interviews.....	35
3.5.3. Document Analysis	36
3.5.4. Data Collection Method in First Case Study.....	37
3.6. Summary	37

Chapter 4: Emerging Themes	39
4.1. “What will the system do?” -The problem of Aligning Objectives.....	39
4.2. Roles and Responsibilities	44
4.3. Data Consistency and Integrity in E-Science.....	47
4.4. Summary	51
Chapter 5: First Year Progress and Proposed Work Plan	53
5.1. First Year Progress.....	53
5.2. Proposed Work for second and third year.....	54
Chapter 6: Summary	55
Bibliography.....	58

Chapter 1: Introduction

1.1. Background – What are e-science systems?

“Each of the [e-Science] projects had great aspirations for bringing about serious change in the world of science and engineering by facilitating new approaches to data analysis, the extraction of knowledge from very disparate sources, and the application to experiments that were until now unimaginable. The future of large-scale, collaborative science is here now ...”

EPSRC report on e-Science (2004)

E-science has been hailed as a new way of doing science. It aims to draw together a number of tools (namely, grid middleware) to allow science to be done in a distributed way (e.g. over vast distances), through using large amounts of computation power and with seemingly endless data storage facilities.

This has a number of implications. In the first instance, it will allow science to be done in a more distributed way. It aims to provide an infrastructure for geographically dispersed scientists to collaborate through the use of a toolkit. Secondly, it will enable the development of large databases or sources of data, whose cost effective storage would have been previously impossible. Thirdly, scientists will be able to gain access to a large amount of computation power through a distributed network of computers. With the use of middleware, the intention is that the above will be seamless in use. The grid is regarded as a key enabler for this change into e-science.

The vision of e-science presents a number of socio-technical challenges. With building middleware, the difficulty lies in the building of technologies which will work seamlessly throughout a number of software/hardware combinations. Grid technologies exist as a family of technologies of which numerous combinations is possible. With data sharing, there is the question of information governance and etiquettes of information sharing at play – i.e. are there ways in which people orient to in order to allow others to build up the same shared understanding of the

information shared (or given) to them? (Hartswood et al, 2005) In terms of IPR, it is a tricky issue in regards to who owns the rights to which part of the data/knowledge produced through collaborative work between two parties (Hinds et al, 2005). The use of multi-disciplinary teams in the building of a single system creates a need for the team to have a shared vision. This, in itself raises questions of how decisions are made during development? How are the shared aims of the system built up? If not, how do they reconcile with these differences? Also, how is a problem framed by each member of the team, each with different competencies and backgrounds? There are a seemingly large number of complex matters to contend with during the building of an e-science system.

By understanding these issues, we will begin to unpack how an e-science system is put together – how it is created and made to work. Making the system work enables us to examine the ways in which dependability is built into the system. By this, we not only mean how developers design the system in order to ensure the continuous running of the system (e.g. backups of data or backup services/servers) but also the way in which users will fit together and make the system work (see Voss et al, 2000). This way, there exists two ways in which dependability can be built into the system. Firstly, as discussed previously, in building a system, tradeoffs exist. These tradeoffs often impact on the dependability of the system – for example, accessibility of the data and the security of the data. There is also an important point that each system will have its own set of priorities in regards to what their definition of a dependable system is. In medical/healthcare systems, data security is of the highest importance (for patient confidentiality), whereas in physics e-science systems, data security is not considered as important due to the nature of the data produced, and availability and timeliness of computational power would more of a priority for them. Secondly, the everyday work of ensuring that a system is running smoothly, to plan, with the correct data and correct procedures, in itself is an achievement in making a system work. This thesis focuses on the requirements capture process of the development of e-science systems.

1.2. Requirements capture in e-science systems

“Accordingly, requirements engineering denotes a set of methodologies and procedures used to gather, analyze, and produce a requirements specification for a proposed new system or a change in an existing system.”

Bergman, King and Lyytinen (2002: 39)

Requirements capture is the process which outlines and captures the specifications, conditions and expected functions of a system. There are numerous types of requirements and specifications, from user requirements which specify what the user wants, to functional requirements which specify what exactly the system will do. However, requirements capture, although it has been considered as a purely “technical” exercise in the past (Bergman et al, 2002), should not be treated as such. A list of requirements is formed under social, political, organisational as well as technical constraints. That is, a requirement is construed as a socio-technical artefact, where its construction, realisation and assessment are shaped by socio-technical considerations. For example, whether a requirement is considered as fulfilled depends on the assessor’s definition of a satisfactory solution of the problem. As Woolgar (1996: 90) points out:

“technology can be regarded as “congealed social relations” – a frozen assemblage of the practices, assumptions, beliefs, language and other factors involved in its design and manufacture”

During the requirements capture process, values and assumptions about the user, the current and future environments which the system will work under are all made (whether explicitly or not is another matter). The main argument here is to view the requirements capture process as a socio-technical negotiation process. In this thesis, I want to examine the issues that arise in the requirements capture process and whether there are certain characteristics that e-science brings which makes it different from traditional requirements capture processes. For example, with the use of multi-disciplinary teams, does the problem of creating a shared vision of the system require more effort than before, or would need the use of new tools? Since e-science has been hailed as a redefinition of science, how much of the values, objectives and assumptions of “traditional” science carry over to e-science? It is these issues that I want to examine.

1.3. Research Questions

The aim of this study is to understand the social, political and technical issues that occurs during requirements capture of an e-science system. These issues can be on a number of levels, whether it would be at an organisational, team or individual level.

Therefore, the central research question to the work is:

What requirements capture issues arise in the system development of e-science systems?

I consider requirements capture to encompass many problems that surround the work of collecting specifications of the system. It presents a number of themes. These could include:

- **Roles and Responsibilities.** Collecting of specifications is no small task. Through the specifications, the functions and expectations of the system is expressed. With such an important task, whose responsibility is it to collect the specifications? Who has the final decision on whether that list of specifications is correct? Are the responsibilities devolved? With e-Science, the organisational boundaries and roles and responsibilities are more blurred than traditional systems development, will this affect the process? Also, e-science brings together different line management structures (mainly due to the heavy role of universities in realising the vision), how will this affect these responsibilities as well?
- **Priorities in Work.** In the work of realising of specifications/requirements (i.e. building the system), what are seen as the priorities for the developers? Which part of the system is given high priorities and why? Is there a particular pattern/reason to prioritisation of work in e-science projects?
- **Managing Expectations.** With the “hype” of e-science, clinicians and researchers are reportedly excited about the possibilities that e-science (and health grids) could afford. However, on an everyday level, how are these

expectations managed between developers and end users? What are the expectations of the system at the start and what did it deliver by the end of the initial funding period? In regards to automation and computerisation, what parts of the system did end users expect to be automated and how did this translate into what can and what cannot be automated? What are the expectations of the scientists and what are the expectations of developers? It is about the creation and formulation of solvable problems.

- **Data Sharing, Consistency, Integrity.** Data sharing is one of the major parts in the vision of e-science. However, there are many dimensions to sharing data, whether it is in term of etiquettes of how the data should be collected or shared. However, data sharing is only one part of the challenges that e-science projects have to deal with. More upstream activities such as how the data is collected also have serious implications. For example, in order to compare data, the way samples are collected and analysed must be consistent manner. If not, it is considered inappropriate to compare them, therefore, how to developers contend with this issue.
- **Rates of Change.** In the vision of e-science, it incorporates the idea of flexibility in terms of the ability to rapidly move to different research areas. However, whilst flexibility is highly desirable, how can developers account and built this flexibility at the start of the system? That is, how is this flexibility turned into requirements? In addition, despite systems always being in a state of flux, they must also have some stability in order to function. How is this “dynamic stability” managed? How often can systems be upgraded/changed? Is it a series of jumps or more gradual evolution of practices?

1.4. The Approach – Empirical (Observational) research of System Developers

Requirements capture is a complex process with many facets. The process has concerns which cannot be drawn out with the simple one time capture interviews or the simple questionnaire. Rather, these processes have to be studied with an understanding to the system's context – the place, the politics, the people etc. It is only through a combination of rich observational and interview data that these interplays can be drawn out.

This approach is not new as a number of researchers have called for more empirical evidence within the field of software engineering. Monterio and Svanaes (1993), argue that the non-technical issues are those that create the largest source of complexity. They claim that to examine these problems would require the examination of systems developers work in real world contexts. Only by examining and understanding real world situations, can we understand better political and ethical problems which can only surface in “real life” situations. Brooks (1990) also highlights this. He points to the complex negotiations that go on in software design, as the “the requirement as presented to the software designer or programmer are usually incomplete and imprecise” (p242). Because of this impreciseness, there lies the need to examine how software developers manage their work and solve problems in order to understand what goes on in the software development process.

Although this view has been incorporated into information systems literature, it is this type of embedded approach to understanding the requirements capture process that has been lacking in the software engineering literature. I will discuss this in greater depth in the Research Methodology chapter (chapter 3).

1.5. Organisation of the Proposal

This research proposal is split into several parts. Chapter 2 will introduce the background and relevant literature to the topic. The background literature will explain the term “e-science”, outline its vision and discuss the types of technologies it will need and the challenges it will face. I will then outline some of literature which view requirements capture as a socio-technical process. Chapter 3 will outline the research methodology proposed. The need to conduct a qualitative study is discussed. It will

also outline the background to the first case study conducted and the data collection methods that I have been (and will be) using. Chapter 4 presents some of the emerging themes from the first case study of the thesis. The first case study has been a 9 month observational study on the development of an IT infrastructure of a translational (cancer) research project. Chapter 5 presents the progress made in first year of study, and will outline the work plan for the second and third years of study. Finally, Chapter 6 summarises the entire research proposal.

Chapter 2: Requirement Capture in E-Science Systems

2.1. Introduction

This chapter introduces the literature from which the thesis is based upon. It is split into two parts. The first covers the definition of e-science, the e-science vision, and how that is being achieved. I also place this in the wider context of the national e-science programme in the UK. The second part covers more specifically to the problems that have been highlighted in literature on the problems of requirements capture in e-science systems.

2.2. E-Science

2.2.1. E-Science – The Vision

“e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it.”

John Taylor
Director General of Research Councils, Office of Science and Technology, 2001

E-science is a vision for the next generation of science which encompasses, global collaborations, distributed networks and large scale science (Atkinson et al, 2002). It is a vision which hopes to bring together technologies that can solve problems that would have been previously viewed as too resource intensive. The main aims of e-science are to provide a flexible architecture where services/technologies can be quickly assembled; the networking of these technologies are (or can be) on a global, distributed scale; and having access to a larger pool of resources. It is about making it easier to interact and communicate between scientists and to have easier access to resources (Fox and Walker, 2003).

The family of technologies that have been developed to realise the vision are Grid technologies. The idea of the grid itself is a grand vision – the ability to gain seamless access to a large amount of computational power and data storage through the networking of lots of smaller computers (cf. supercomputers). If configured rightly, the utilisation lots of computers with a smaller computation power would be more powerful than having one supercomputer. The most notable example of a grid like application is the SETI@home program, where users can download an application from its website that has a chunk of data to be processed. The processing takes place whilst the computer is idle, and the results are then sent back to the main server. Note the distributed and divide and conquer approach to this. Here, are two challenges which present itself in that grid vision. First is the challenge of networking hundreds, or thousands of computers, each with a unique combination of hardware and software. The second is the ability to access these resources in a seamless fashion (i.e. on demand computing where the users do not need to worry about configuring the application).

The challenge of building Grid technologies is immense. This has lead to the creation of the semantic grid and service orientated architectures. The semantic grid is a vision created by combining the semantic web and the grid. It is characterised by “an open system, with a high degree of automation, that supports flexible collaboration and computation on a global scale” (DeRoure et al., 2001). Very simply, it is the building of grid technologies that will enable data to be repurposed and linked together in a flexible way. Service Oriented Architectures (S.O.A.s) are also similar, in that entities (such as service providers) provide a particular service (e.g. computation power or data resources) by agreement of a particular contract (DeRoure et al., 2001). Under S.O.A.s, services or resources can be quickly assembled in order to fulfil some need.

Note that it is not only the computing infrastructure that has to be built in order to support this, but also the training that is involved in teaching scientists which problems are applicable to use of e-science tools. For example, simulations and modelling are often good applications for e-science since problems can be ran in parallel to each other. However, other applications which involve sequential processing of data might not be as applicable. Fundamentally, it changes the way that

science is done. It is hoped that with the use of computing tools that it will enhance and support the solving of bigger scientific challenges (DeRoure et al., 2001). It is this change that will spark off changes to the bigger scientific community.

To help realise this vision, in Nov 2000 the UK government announced funding of £98 million for an e-science programme, led by programmes from each of the research councils. The programme is managed by the EPSRC on behalf of the funding council and a steering committee chaired by Prof David Wallace with Prof Tony Hey as the Director of the E-Science programme. The National E-Science Centre (NeSC) arrived soon after and a number of e-science nodes have appeared across the country. Soon after, in 2004, the National E-Social Science centre was opened. The e-Science programme is now in its second phase of funding.

The notion of e-science has been applied to several domains with a number of demonstrator projects under way (and completed). With its “promise” of availability of large computation power, e-science has been applied to computationally intensive domains such as astrophysics and micro-biology. For example, AstroGRID (<http://www.astrogrid.org>) is a long term project which aims to create an open source “virtual observatory” environment for astrophysicists on a global level. The project includes the building of a federated database, alongside data mining tools. It also allows users to upload and run their own algorithms on the data available. Medical researchers and clinicians can also benefit from e-science tools. For example, the eDiaMoND project is a large scale project which was aimed to demonstrate the use of grid tools in the National Health Service (NHS) (Jirotko et al., 2005). It contained a federated database of mammograms spread over 12 sites. It was a project which would allow radiologists to search and compare similar mammograms from across the other sites – a new resource to radiologists. They could acquire and query particular images from the database of mammograms (ibid).

2.2.2. What challenges does it face?

The development of e-science systems is inherently difficult; both due to the lack of maturity of the technologies and with the number of (organizational) boundaries

involved. It is because e-science is distributed across “traditional” organizational boundaries such as particular departments, universities, or companies, that there is heavy emphasis on the people, power and trust involved. The assemblage of components not only revolves around the hardware, software, people within one place or with a common goal, but rather, a number of different actors which might not actually meet physically. However, because e-science is based upon distributed networks, e-science systems tend to be more dependent on other systems. The negotiation process goes beyond the end-users and the developers, or between management and shop floor workers – rather the development of the system will depend largely on the trust and relationship with other systems.

The move towards e-science has presented many challenges – both technically (e.g. tools and the building of middleware) and socially (e.g. sharing of data (Hartswood et al, 2005), argument of intellectual property rights (Hinds et al, 2005), organizational boundaries). Recently, the focus has been on the technical challenges of e-science systems. After all, building technologies which will work seamlessly throughout a number of hardware/software combinations is not a simple task. Whilst these provide knowledge and experience on the practical issues that need to be addressed in the building of e-science systems, I argue there has been insignificant focus on the process of developing of e-science systems itself.

2.2.3. Characteristics of E-Science Systems

With the complexities that arise from an e-science system (especially its distributed and collaborative features), it is logical to ask the question – Does the development process of e-science systems differ from traditional systems? In turn, does that mean the requirements capture process for e-science systems are different too? i.e. Do e-science projects make certain problems in systems development more prevalent? Does it require extra care in addressing these problems? By trying to understand whether there are differences, and the particular characteristics of e-science systems, we will better understand the consequences for requirements capture of e-science systems.

In their paper, Momtahan and Martin (2002) draws from their experience was working on the building the European Union DataGrid project whose aim was to build “higher collective services” such as workload and data management tools. They explore the extent to which current software methodologies are appropriate for the development of e-science tools. They also discuss whether there are needs for different models of development.

They outline six software engineering challenges of e-science systems gained from their experience on the EU DataGrid project:

- **Volatility of requirements.** Requirements are difficult since each group of scientists will have their own sets of requirements (e.g. physicists and medical researchers); grid technologies are relative novel and there is no set paradigm for them and also, it is only when the system is used that problems will arise and the required specifications actually emerge.
- **Geographical separation and communication.** Production of software through geographically dispersed developers is much more difficult. It might create the need for more documentation, and hence, larger amount of time devoted to administrative duties and creating a shared vision/meaning.
- **System decomposition.** Breaking down the system into components is much more difficult since the developers are separated. Also, because the system is developed with a number of partners, each party will have their own agenda which might be in conflict with others.
- **Project processes and authority.** Individual projects sites liked to work according to their own working practices and project processes. Each site (organisation) managed their own work, and largely ignored an overall plan. Therefore, consistency was lacking in managing particular issues.
- **Project planning and tracking.** Whilst industrial projects often recruit/enlist more resources as the project goes on, academic based projects (which is how

e-science is largely driven) has got a specific flat resource profile, i.e. developers are employed for a specific amount of time, and staff can only be allocated projects in a sequential manner.

- **Morale, Attrition and culture.** Academics are generally paid less than their industrial counterparts, and certain staff are lumbered with responsibility in which they have no authority over. All these contribute to low morale on the project.

They go on to outline the implications for software production:

- **Large geographical distribution.** With the developers being distributed, the communications overhead to coordinate the work increases.
- **Absence of an authoritative management structure.** Because of the largely academic and distributed workforce, there was a lack of an authoritative “project manager” figure.
- **Requirement from different stakeholders with different perspectives.** Different stakeholders who will have different rewards, gains and motives for doing particular things on the project. Also, with the different number of stakeholders, the number of requirements will be large as well, because each stakeholder will have different needs.
- **Evolving requirements and design due to the novelty of the software.** Requirements will evolve over time and this will be impacted by the software used. Due to the innovativeness of the software, users will find new ways of utilising it and as a result, produce more requirements.
- **“Square pulse” human resource allocation.** As mentioned previously, academe tends to fund projects with a relatively flat resource profile, i.e. human resources are allocated a particular time on a project, rather than be

“pulled in” whenever necessary – a practice much more common in industrial projects.

- **Academic software development practices.** Academic generally has no universal software practices/methodology (cf. industrial practice).

Momtahan and Martin (2002) concludes that the model of development for e-science projects are more akin to open source models of development rather than traditional “industrial” models. The open source model of development is one which is characterised by distributed collaboration of writing software, with minimal working prototypes in order to gain as quick and much feedback as soon as possible. It has been described as an “unusually rapid and iterative form of Boehm’s spiral model” (ibid). There are similarities between the two – for example, both require peer review.

However, Momtahan and Martin do not further elaborate on commonalities of the two and does not expand on the implications of such work. Their argument, of course, is debateable since the project was based upon open collaboration in the development of a sustainable software tool for data processing. Not all e-science projects share these characteristics. For example, in medical research, this might be different. This is especially in relation to projects which work with patient data and therefore there might be more reluctance to share the knowledge involved. The importance of patient confidentiality might also limit the number and choice of collaborators with these projects. Also, the scale of the system will impact on whether the project would evolve/exist similarly to an open source project. Whilst their comparison of development in e-science systems to an open source model is debateable, they nevertheless highlight (even if tentatively) the differences between the “traditional” software engineering models of work and those present in e-science systems.

2.3. Requirements Capture

“More than half the cost of the development of complex computer-based information systems (IS) is attributable to decisions made in the upstream portion of the software development process; namely, requirements specification and design”

Walz, Elam and Curtis (1993:63)

Requirements are the list of properties in which the system has to meet. It states what the system should do, and under what circumstances and situations it should work under. The system normally works in order to solve some problem or to satisfy some need. Requirements capture is the process in which these properties and constraints are operationalised or made explicit in some form (Sommerville, 2001: 98).

Whilst it may seem easy at first to define the properties a system should have, requirements capture is not as simple as it seems. As Walz et al (1993) states in the above quote, the impact of the decisions made in the “upstream processes” such as requirements gathering and capture is immense. This is mainly because particular decisions are built upon other decisions and as a result (Button and Sharrock, 1996), if the first few fundamental specifications/decisions were based upon inaccurate assumptions then effort needed to rectify those decisions is considerable. It is for this reason that a large body of literature have grown around the topic of requirements gathering and specifications.

In traditional software development models such as the waterfall model or the spiral model, requirements gathering always occur at the start of a project. The requirements specify the scope and functions of a system. It defines how big the system is, what the main use of the system is, what the users of the system expects, what the conditions the system should work under. The list is endless. From the requirements specification document, the system is built. This document is important for two reasons. The first is that it serves as an agreement between the developers and the end users as to what the users want and what the developers will build. It serves as a common, joint vision/idea of what the system will do and what it will be like. The second is that it allows the developers to understand what they have to build and have a list of conditions in which the system has to satisfy. Depending on the type of software development model the project adopts, will determine whether the specifications are revisited and how often they will be revisited.

However, defining what a system should do is a complex task. Different users of the system (whether on a group or individual level) might have different perceptions of

the systems function and scope. As a result, there might be specifications might be in conflict with each other (Hertzum, 2004). This is especially the case for large scale systems which involve more than one group of users. This is a commonly identified problem within systems development. For example, the end users of a system might want flexibility in entering text, however, this would make searching text (i.e. the secondary users of the data) much more difficult. It is these competing and sometimes conflicting problems in specifications that developers are faced with.

Also, specifications may change over time i.e. end users might change their idea of the system as they are asked for specifics. As time goes on, and the end users have had a chance in understanding their problem better, they might change their conception of what the system will do, or should do. However, end users rarely understand the consequences of such changes:

“Customers rarely understood the complexity of the development process and often requested frequent changes to the requirements. They underestimated the effort required to re-engineer the software, especially when the system involved tight timing or storage constraints. They rarely understood the impacts that rippled through the software when changes were made and the coordination required to document and test these changes. As a result, customers could not understand why changes to the requirements were so costly”

Curtis, Krasner and Iscoe (1988:1276)

There may be specifications which could be very difficult to make a decision over. Uncertainty and fluctuations surround requirement specifications.

Sometimes, the end users might not even know or understand what it is that they want (Hertzum, 2004). The literature of ethnomethodologically informed ethnography, participatory design and co-realisation has borne out of this problem of just understanding user requirements (see Asaro, 2001; Mumford, 1987; Dourish and Button, 1998; Hartswood et al, 2002). Ethnomethodologically informed ethnography is used to help designers to understand how users currently work. These working practices and the implicit assumptions that users have in their workplace are used to inform the specifications of the system. In participatory design, users are asked by the designers to re-specify their work in a structured way through formal sessions (such as meetings) between developers and users.

At the same time, it should be emphasised that the requirements specifications should not be seen as something that is teased out of the end users or the environment around them. Rather, “requirements are not waiting to be discovered by an analyst, but rather they are systematically constructed through an iterative and evolutionary process of defining and redefining the problem and solution spaces” (Bergman, King and Lyytinen, 2002: 48).

Even when there are clear requirements, these can be open to interpretation by each developer (Curtis, Krasner and Iscoe, 1988). They are open to interpretation depending on the constraints (and the uncertainty and ambiguities of such constraints) placed upon the developer (Bergman et al., 2002). For example, the specification of having the ability to enter free text could mean having a small text box of around 255 characters or it mean having an option of attaching a text file. In other words, the criteria by which a specification is successfully met is open to interpretation (Guidon, 1991). It is this openness that is a problem. Although it has to be recognised that some specifications are more open than others. Whilst a potential answer to this problem is to say that specifications have to be as concrete and grounded as possible, with as many operational details, this is merely a gloss over a) the types of socio-technical problems that has to be tackled and b) the complexities in which will occur during systems development. To implement these specifications would mean making assumptions about the usage of the system and how it will evolve (Akrich, 1992; Woolgar, 1996; Winner, 1987). I agree with Bergman et al. (2002) conceptualisation of technology as “heterogeneous engineering”. As they point out, mainstream requirements engineering literature makes false assumptions that technical problems can be divorced from social problems, and as a result, be distilled into a document:

“This is because they are based on a fallacious assumption that business problems and political problems can be separated from technical requirements engineering concerns of how to specify a consistent and complete technical solution to a business problem. In contrast, large scale system development initiatives involve a simultaneous consideration of business, institutional, political, technical and behavioural issues.”

Bergman et al. (2002: 39)

The system produced at the end of the development process is hence a socio-technical “ensemble” (Law, 1987). Bergman et al (2002) term the possibilities of which a satisfactory solution can occur as the “solution space” of the system. Most solution spaces are large due to the different combinations of socio-technical problems that can occur.

Not only can the space of specifications and requirements be large, this space will evolve over time as well. Specifications constantly change throughout the life of a project. One explanation is that specifications are inherently difficult to make concrete (Button and Sharrock, 1994). Another reason could be through changes in the external environment or the understanding of the problem being improved (Curtis et al., 1988; Button and Sharrock, 1998). This, of course, have resonance to the e-science systems, as the vision of rapid moving, globally distributed science is brought to fruition. The realisation of such flexible systems will result in fluctuating requirements from the start of the e-science system and all the way to the end of its lifecycle.

One way to conceptualise the requirement process is as a learning process (Ronkko et al., 2005) or as a knowledge transfer process (Guidon, 1991; Curtis et al., 1988). New knowledge (and hence learning) can come from an improved understanding can come from knowledge of the application domain (ibid).

“Some performance differences were determined by how deeply programmers understood the application for which they were writing programs. Specification mistakes often occurred when designers did not have sufficient application knowledge to interpret the customer’s intentions from the requirements statement. Customer representatives and system engineers complained that implementations had to be changed because development teams had misconceptions of the application domain”

Curtis, Krasner and Iscoe (ibid: 1271)

Curtis, Krasner and Iscoe (ibid) attributes a lack of understanding of the end user’s application domain as one of the major reasons why systems developers do not understand the requirements provided to them. This misunderstanding results in uncertain and fluctuating specifications as the requirements capture process is misunderstood. In their study, Curtis et al (ibid) observed a team developing a system

in a research project and the way in which developers shared project relevant information with each other. The application domain was novel and was unfamiliar to the developers (a situation that developers of e-science systems will be placed under). Their study had several interesting findings. Team members exchanged information through discussions. One developer would declare their “statement of position”, this viewpoint facilitated discussion between the other developers, the result of which meant that some knowledge was shared to the other developer (which might mean changing the other developers “beliefs”). “Exceptional designers” were described as interdisciplinary as they would incorporate several different bodies of knowledge (Curtis et al., *ibid*: 1272). This meant that to be effective designers, they not only had to understand their own domain of knowledge, but to be able to assimilate other areas of knowledge and effectively incorporate them into the system. This is also a point very relevant to the development of e-science systems, as the development of which usually involve multidisciplinary teams. It is this interaction with other experts in the team which provides this knowledge transfer (Ronkko et al., 2005). In addition, a considerable amount of time was spent on “coordinating a common understanding among the staff of both the application domain and of how the system should perform within it” (Curtis et al., 1988: 1275).

Although the knowledge transfer has been described in the context of the wider systems development process rather than within requirements capture, I argue that a lot of the principles are applicable. For example, in requirements capture process, developers have to decide on a common, shared vision. This can only be done by transferring the knowledge of their domain and absorb the views of the other developers in the team. By understanding the process in which requirements are gathered, implemented and played out in the arena of systems development, we can better understand how to better support the process (Walz, Elam and Curtis, 1993). It is only by understanding the complex ways in which organisational, technical, social and political ways in which requirements are formed and built into systems that we can better manage the requirements capture process (Bergman, King and Lyytinen, 2002).

2.4. Summary

In summary, this chapter has presented two bodies of literature – e-science and requirements capture. The first half outlined the vision of e-science in the U.K. The e-science vision brings together many experts across a number of domains developing and sharing resources together (such as data, computation power). As a result, e-science systems are complex entities.

In the second half, I argue that the requirements capture process should be more than a technical process. There are many complex issues that need to be understood. It is important to recognise that specifications are difficult to manage, as its definition, implementation and evaluation are often subjective and open to interpretation. Also, I argue that the requirements capture process should be seen as a learning/knowledge transfer process. This conceptualisation is useful as we consider (in chapter 4) the ways that developers have to align their objectives especially working in a multidisciplinary team.

Chapter 3: Research Methodology

3.1. Introduction

This chapter outlines the proposed research methodology for this thesis. This covers from the reason for a qualitative approach to the study, to the case study selection and through to the methods of data collection and analysis. Research methods are highly important, and as it will impact on the result of the research, and therefore, must be carefully chosen.

This chapter is split into three sections. In the first section, I explain the reason for conducting a qualitative study, and the need to gain rich, descriptive data to understand the issues that were outlined in the introductory chapter. The second section discusses the choice of using case studies, what case studies were chosen, as well as the background to the first case study that I have been researching on. This leads us onto the third section which discusses the methods of data collection in this thesis. The case for using mixed methods is made. The types of include observational methods, interviews and looking through physical and electronic documents. Through a combination of these data collection methods, the complex environment of a systems development project can be uncovered.

3.2. The need for a Qualitative Approach

“There is growing recognition that research on how teams actually go about making requirements determinations and design decisions can provide valuable insights for improving the quality and productivity of large-scale computer-based IS development efforts. Traditional models of group dynamics, group decision making, and group development are not rich enough to thoroughly explain the real-world complexities faced by software design teams”

Walz, Elam and Curtis (1993:63)

Systems development is a complex process. A wide range of factors can affect the requirements. Designing systems creates a number of tensions that may or may not be easily observed. The need to understand and examine the real world work of developers requires an understanding of the real world contingencies that arise during this process (Monterio and Svanaes, 1993). This includes the political pressures that developers are placed under in the workplace. There are deadlines and negotiation processes which are not visible on the surface. People have different viewpoints, problems can be exaggerated and tensions that arise are all part of the rich backdrop that systems development exists in. All these constraints and pressures in the workplace will shape the final socio-technical artefact that is produced (see Schwartz Cowan, 1985 or Fallows, 1985 for example of technologies shaped by external factors). In systems development, these pressures are present throughout the working life of the system. To uncover these pressures is to truly understand how the requirements were shaped, negotiated and considered achieved.

With the need to examine these political, social, organisational constraints, it needs to be recognised that these issues are often difficult to uncover, due to the implicitness of their nature and the politically sensitive environment which they exist in. The important point here is that rich, thick descriptive data is needed in order to understand these issues. It is for this reason in which a qualitative approach will be undertaken in this thesis. Quantitative methods are not suitable in this case as it will not capture the thick, rich description of the situations that arise will provide the background, and meaning to the resolutions that is needed in order to understand fully why the developer did what they did. Placing a numeric, quantifiable measure on the factors discussed here would imply there is a simple cause and effect relationship which does not occur in the complexities of the real world.

However, the major problem with qualitative approaches is that it is time and resource intensive. To collect and analyse the data requires a large investment of time. It requires a significant amount of time to conduct an observational study, both at the field site (observing) or away from it (writing up the fieldnotes). It also requires a significant amount of time to conduct interviews, especially the semi-structured or

unstructured interviews (Data Collection Methods – Section 3.4). Qualitative data are notoriously time intensive to codify and to analyse.

3.3. Selecting Case Studies

The selection of case studies is very important. The case studies reflect the empirical evidence produced to answer the research questions. Therefore, the cases should be selected so that they are the most appropriate to answer the research question asked. For a start, there is which sector of e-science the projects should be from. Then there are questions of how many to select, and also the maturity of the project. There are a number of practical considerations too. For example, access to the case study, and time and resources to carry out qualitative research, which is often time and resource intensive. These are highly important questions that are shaped by the research question as well as affecting the research itself.

One of the problems with identifying which projects to select is that the number of e-science projects are diverse. They range from long term projects in astronomy (AstroGRID) to relatively short term projects in digital mammography (e-Diamond). There are a number of differing characteristics which includes:

- **Time span** – ranges from two years to long term resource
- **Geography** – could be centralised or geographically distributed
- **Subject/Science area** – from astrophysics to medicine
- **Computationally intensive or Data intensive** – whether they rely heavily on computational power or heavily on data storage
- **Maturity** – completed, in the process of development or only started development
- **Middleware development or specific system/tool development** – whether the project is to build generic grid tools or develop specific tools for one specific project
- **Number of collaborations** – the number of collaborative partners there are in the e-science project

- **Utilisation of bleeding edge technologies** – how innovative is the technology used
- **Open or closed** – whether it seeks contribution from a community or just a selected project team
- **Project size** – the number of people working on the project

The above list is not exhaustive. It is easy to see how the selection of e-science projects could be difficult with the number of characteristics that can differ. As stated in chapter 1, the research question revolves around requirements capture and the issues and tensions that arise revolving requirements in e-science projects. For this purpose, the case studies of e-science projects should be diverse in order not to be caught up with the characteristics of one type of project. For example, data protection and patient confidentiality is highly valued in the medical research arena, but such problems do not arise in the particle physics projects where the data are not produced from people. However, there is a danger that studying projects that are very diverse would be difficult to generalise across; especially since the number of case studies undertaken would be unlikely to reach over 3 or 4 (explained below). Case study selection will also depend on access and other practical issues such as geographic distance.

The reason for only examining 3 or 4 case studies (at the most) is due to time and resources. The type of qualitative research discussed here would require a significant amount of time to collect and analyse the data, it means that the number of projects covered would be low; however, the data gathered would be very rich.

In addition, the case studies should cover projects that are in different stages. Observational work started in the first case study whilst it was the system was in its infancy stage. It is envisaged that the second case study would be a completed project (or a long term ongoing project) in which result of the requirements capture process can be assessed after the project is completed; and therefore, the project members can reflect upon the requirements capture and how it affected the project's life.

Currently, only one case study has been undertaken and chosen. For the reasons I have mentioned above, it is not an easy task to choose another two case studies and this will require more planning and access negotiation. The next section outlines the background to the first case study, and also outlining the plans for the second and third case studies.

3.4. Research Case Studies

Three case studies are planned to be carried out. The first is the development of an IT infrastructure for translational research based at a city hospital. This case study was chosen as it provided an excellent opportunity to study the development of an e-science system from its relative infancy. It provided a good location for rich, ethnographic/observational data due to its central location and the relatively small development team. The project has been running for 18 months, and had a challenging remit of linking medical data, which, traditionally has been seen as a difficult task in political and organisational terms. In addition, the relative ease in gaining access to the case study was also a factor. I will outline the first case study in greater detail below. The second and third case studies are to be decided, and plans for these are outlined below.

3.4.1. Case Study 1

Background and aims of TransProject

The first study is TransProject, which is the building of the IT infrastructure for translational research in a city hospital. TransProject is part of a network of translational cancer research centres across the U.K. The vision of Translational Research is to quicken the process between basic bench science and the delivery of treatment to cancer patients, and back. Each centre is linked in a network with each centre being autonomous in their work. TransProject aims to do this through the building of an information system to facilitate cancer research. The system will cover from the nurses obtaining consent from the patients to the tissue being analysed through Proteomic and Genetic data.

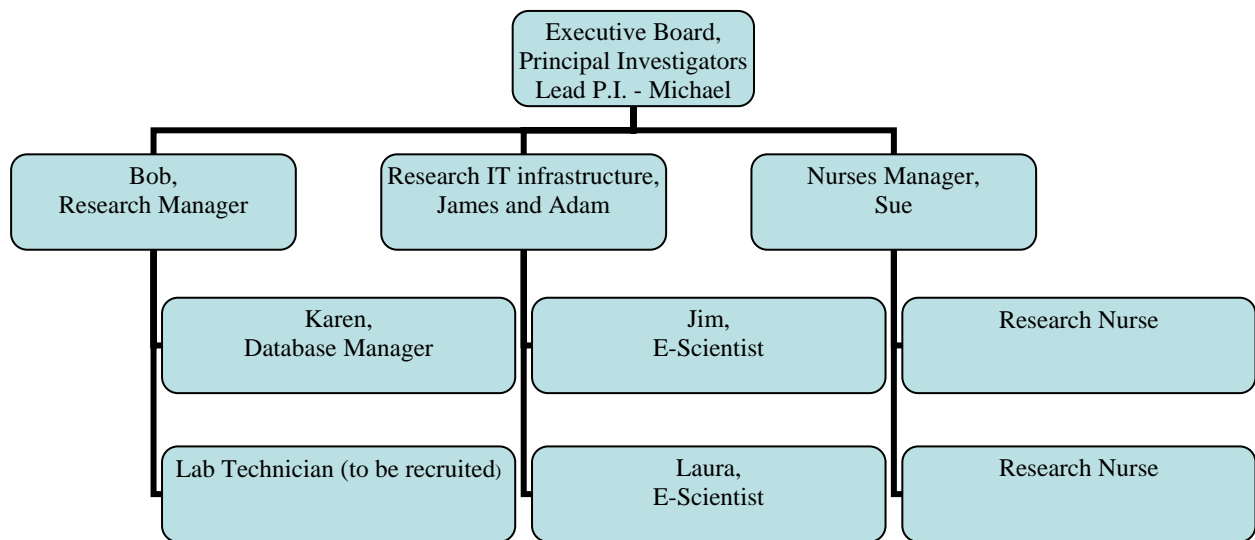
The aim of TransProject is to build an infrastructure in that facilitates the recruitment of patients into cancer research. The infrastructure will need to support all parts of the research process – which includes epidemiology studies (factor x is significant in determining whether patient y will develop this type of cancer) and clinical trials (testing of new treatment regimes upon patients). The infrastructure consists of systems and practices which should support the collection of the core dataset, data linkage with other sources (such as death records), data curation and its analysis.

On a basic level, TransProject will have two nurses who recruit cancer patients into the program. Tissue samples are routinely taken as part of the treatment of patients and for consenting patients, parts of these samples will be taken for research purposes. In addition to this, blood will be taken for DNA analysis. Other data such as food frequency questionnaires, lifestyle questionnaires as well as the family history will be recorded in the TransProject database. Data from patients' from clinical records will be acquired by record linking with existing data sources such as disease registers or audit data. One outcome from this is that with TransProject, datasets can be utilised for cancer research without the need for each individual cancer study to recruit patients. In summary, TransProject is an attempt to build a common platform for different areas of research (e.g. Colorectal Cancer, Breast Cancer) as well as different stages of research (e.g. epidemiology studies, clinical trials).

The centre is located in the Clinical Research Facility of a large city hospital. The work done by TransProject will be centralised, local based work at the hospital, although it is envisaged that the centre will grow to encompass both more hospitals and more types of cancers.

Prior to TransProject, there was a study carried out by the same Principal Investigator called CancerStudy, which was a nation-wide project, but only for one type of cancer.

Organisational Structure



The TransProject team is organised into four main sections. The top level is the executive board of Principal Investigators (P.I.s) which oversees the direction of the project. Bob, as the research manager, is the “project manager” of the entire project. He is responsible for the day-to-day running of the TransProject progress. He is also the line manager to the database manager and the lab technician. The database manager is responsible for the curation (i.e. collection and accuracy) of the database. The lab technician will be responsible for preparing and storing tissue and blood samples. The Research IT infrastructure team is headed by the head of two local research groups, one based in informatics at a local university and another based at a genomic unit at another local hospital. An e-scientist was recruited from each of these groups. Jim is responsible for the record linkage and building of the IT infrastructure. Laura, on the other hand, is responsible for creating the link between the genomic facility and the TransProject system. She will also be partly building the TransProject system too. The final members of the team are the two research nurses, who’s responsibility it is to recruit cancer patients from the ward and to enter patient data into the TransProject system. The nurses are managed by the local nurses manager at the Clinical Research Facility of which the TransProject team is located. Just to clarify, when I refer to the development team, I include the Research Manager,

Database Manager, Lab Technician (although yet to be recruited) and the two e-scientists. They are the members of the project which will help to develop the entire system – not just the IT infrastructure, but the entire system which includes the S.O.P.s.

In terms of research, since the development team is quite small (5 to date) with only 3 members working full time on it – another 2 on part time arrangements – more emphasis was placed on understanding when would be the most appropriate time to be “around”. As the Research Manager has a separate office, it is possible for little “action” to note around the office during this period of time. In addition to this, two research nurses will be recruited around Dec 2005/Jan 2006. The last remaining member of staff - a lab technician will be recruited early 2006.

Evolution of the System

Work started on TransProject around July 2004 with the recruitment of Jim, the first e-scientist. At that time, there was a broad remit and specification of what the system was to do. Also, Laura (the other e-scientist) started work on TransProject at the same time through standing in for Adam at various TransProject meetings. I started observation around start of February. At this time, Jim had started to approach CancerStudy about their infrastructure. Jim then, at this point, started writing an applications framework. The applications framework had the ability to generate web based applications relatively quickly through the specifications of some XML documents. He said he was frustrated at the lack of requirements specified and that through building the applications framework, he could quickly deploy any changes in specification that needed to be made.

By March 2005, Karen was recruited as the database manager and Bob was recruited as the Research Manager onto the project. Karen started in March and due to availability, Bob could not start until mid-May. When Karen started as Database manager, discussions revolved around which types of programming languages to use, as Karen would have a large role in the development of the system. Discussions also arose regarding the datasets of the system – i.e. what data TransProject should store and where the sources of that data would be. At this point, there were still great confusion over what data should be stored and the types of data that would be stored.

As a result, Jim and Karen called a day long datasets workshop meeting. Various members were invited to the workshop, including Bob (even though he had not started work), Laura (who presented on the genomic research group system), and Diane (who is the database manager from CancerStudy). Various P.I.s were also invited but did not attend. The results of the workshop was inconclusive mainly due to no P.I.s were there to make a final decision. Instead, the workshop identified some data that should be discussed with the P.I.s. It was not clear how much effect the workshop did, but as a result, Jim and Karen decided to proceed with programming the National Minimal Dataset (NMD) for colonrectal cancer into XML format, which could, in turn, be inputted to the application framework as a template to generate the web application.

Also, around the same time, through discussing TransProject with various other studies and other local cancer research organisations and the clinical research facility itself, TransProject began being involved in a local healthcare IT group which met to discuss the IT infrastructure in the local healthcare region. As a result, they became involved in a clinical research system jointly built between a number of organisations which aims to support the work in clinical trials.

Around mid-May, Bob starts work as Research Manager and soon after, the team had weekly meetings whereby they would update each other on their progress. The reason for this meeting to provide a way to gather everyone on the team to one place which is problematic considering the two e-scientists often worked away from the TransProject office. By late July, Diane started attending the weekly meetings and starting work on the project part time. At this time, there were also discussions of the job being split between Diane and another colleague Tom, who is a statistician. It was hoped that later on in the study, when the IT infrastructure was set up, Tom could help with the data analysis.

Soon after the weekly meetings started, there were discussions that the Chief Scientists Office (CSO) who funds TransProject was planning a site visit in September 2005. As a result, all through July and August there was a visible rush towards getting as much of the system completed. There was clear tension within the weekly meetings as each team member had different priorities for the direction of the

system. There were heated debates about whether the design of working practices or whether user interfaces were a priority. Then Michael, the Lead Principal Investigator started (irregularly) attending the weekly meetings too. This helped the development team to clarify questions (and to a certain extent, the direction) for them. It was around this time that Jim set up several collaborative tools such as a wiki, and a Concurrent Version Server (version and backup tracking software).

It was clear by September that the site visit was not going to occur due to the TransProject Headquarters being disbanded. However, despite the apparent “breathing space”, Bob was actively trying to recruit the two research nurses. As a result, despite the deadline for the CSO visit did not happen, the tension was continuously mounted upon the development team as there was a rush to work to the new deadline of when the nurses would start. The deadline for the nurses moved from start of November, then towards mid-November, then to the start of December. I stopped the first set of fieldwork around the start of November. Just before I finished the first phase of fieldwork, the applications generator was sufficiently built that it produced web applications from the XML files hand coded by Karen.

3.4.2. Other case studies

As mentioned above (section 3.3), selecting case studies are very difficult. It is hoped that access to two further case studies can be gained. It is hoped that the other case studies would both be outside of medical research, and one would also be a completed project, and the other would be an on-going one in order to capture differing attitudes over the course of the projects. It is hoped that the selection and negotiation of access for the second and third case studies would be done in the second year of study.

3.5. Understanding the Real World: Data Collection Methods

There are a number of data methods available in doing qualitative research. Some of these include:

- Questionnaires
- Interviewing
- Participant Observation
- Document Analysis

In this thesis, I will use a combination of these methods. This approach – mixed methods – is used here because of the wide ranging types of case studies there are available. Because at least one case study undertaken will be a project that has been completed. As a result, observational methods would be inappropriate. However, for projects that are on-going, it would be a wasted opportunity to only collect interview data when an opportunity to collect rich data from observational work is available. It is for this reason that a mixed methods approach is undertaken. It should be noted that in an ethnographic study, the use of interviews and documents often be used.

In the next section, I outline observational methods, interviews and documents analysis; plus the data I collected in the first case study.

3.5.1. Observations Methods/Ethnography

“Ethnography is the art and science of describing a group or culture. The description may be of a small tribal group in some exotic land or classroom in middle-class suburbia. The task is much like the one taken on by an investigative reporter, who interviews relevant people, reviews records A key difference between the investigative reporter and the ethnographer, however, is that where the journalist seeks out the unusual – the murder, the plane crash, the bank robbery – the ethnographer writes about the routine, daily lives of people. The more predictable patterns of human thought and behaviour are the focus of inquiry.”

Fetterman (1989: 11)

Observation is a qualitative data collection method in which gathers data by the immersion of the researcher in the situation/environment under study. The researcher aims to gather data from the environment in which no variables can be controlled. This type of research involves the study of the subject under natural conditions.

There are a number of forms in which ethnographic data can be collected in. Traditionally data has been collected in the form of fieldnotes written by the ethnographer. Through pen and paper, many ethnographers encourage writing as much as possible about each subject each day, no matter how small the occurrence. The fieldnotes themselves can be in several forms. For example, Emerson, Fretz and Shaw (1995) suggests that there should be three forms of fieldnotes – observational notes, methodological notes and analysis notes. Observation notes are notes on what happened, recording precisely (as much as possible) about what the ethnographer saw and felt, noting down what was said. Methodological notes are about the ethnographer's way of eliciting information – about which subjects were most informative, and whether there were special ways of getting information (for example, with software developers, by collecting some emails that they have to communicate with each other provide good insight into their work and negotiation of their work). Analytical fieldnotes are notes that provide a quick analysis of the data collected, often immediately after the event.

This account will contain some of the problems/issues encountered, who had what involvement with the project and circumstances in which they occurred. It is only by understanding and “being there” that the full implications can be written down – which both quantitative and other qualitative methods such as semi-structured interview cannot capture. However, there are also problems to using an observational method. One problem of using ethnography is that as D'Adderio (2004) mentions, the work of the individual programmer is very much alone. And sometimes it is difficult to find out what are the rationale behind their work. It is also difficult to constantly ask what they are doing, and if not ask them, then it might be inappropriate to look over their shoulders.

3.5.2. Informal and Formal Interviews

Interviews are a way of eliciting information through getting the informant to talk about a certain subject. Interviews can be both formal and informal, and they can take in the forms of semi-structured or unstructured. In a semi-structured interview the interviewer has a list of questions that he/she wants the interviewee to cover. These

questions are generally used as an aide-memoir, and the order of the questions can be changed in order to adapt to how the interview is progressing. Unstructured interviews have fewer restrictions. There are only a number of topics (in which the interviewer can agree with the interviewee at the start of the interview) and the interviewee can be free to discuss whichever topic, in whichever way they wished. The use of semi-structured and unstructured interviews is common in ethnography, especially to supplement participant observation.

In systems development literature, these two types of interviewing techniques allow the observer to gain a better insight into the work involved in the workplace. The interviewees can explain their rationale for doing certain types of work in a certain way. Structured interviews are generally not used in ethnography – especially in systems development – as these provide few insights into the subtleties of working in the workplace. Apart from semi-structured and unstructured interviews, there is another major distinction between interviews – formal and informal. It is in ethnography that we mainly make this distinction between informal and formal interviewing. In interview based research studies, all interviews are formal based – because they are all scheduled with the interviewee with a pre-agreed time and place, and probably recorded as well. However, informal interviews occur any time, any place during the workplace, and it flows as a conversation.

3.5.3. Document Analysis

Apart from fieldnotes and interviews, another major source of information for the ethnographer is the use of documents (written or electronic). Documents are usually an “official” channel of communication. These could come in the form of system specifications, proposals, forms of communication between team members etc. Documents provide a good viewpoint of who said what and when.

They are also a good source of comparison of versions of events. For example, through looking at documents that developers have produced and comparing them to their accounts of what has happened, it is possible to note the differences in what the “official” version of events are, and what the developers thought.

3.5.4. Data Collection Method in First Case Study

Data collection in the first case study was in the form of fieldnotes through observational methods. Once the regular weekly meetings started, I offered to write up the minutes of the meeting since I was making notes on the meeting itself. It also gave me a “legitimate” reason for being in the meeting itself, whilst levitating the burden of writing minutes for the meeting when there were no “administration” staff available. Methodologically, this was also a way of getting members of the team to be comfortable of having an observer around writing notes of what was said and why. The majority of the data since then has been collection of notes around Tuesday mornings – when the meetings were held and Tuesday afternoons – when all of the team members were in the office and actually interacting with each other. This was crucial with the two e-scientists only being in the TransProject office once per week.

In addition to the fieldnotes taken during the weekly meetings, I also recorded some meetings too (with permission from everyone present). I recorded a total of 8 meetings (out of 16). I also conducted informal non-recorded interviews with various members of the project team. Notes were taken after these interviews and formed part of the fieldnotes taken. Electronic documents which includes electronic mail (email), that I could access to also formed part of the fieldnotes and I collected. Finally, notes made on the wiki also formed part of the understanding of what was going on in the project.

3.6. Summary

This chapter discussed the research process being adopted in this thesis. It outlines the reasoning for conducting a qualitative study. To fully understand the complex relationships that surround the process of requirements capture requires an understanding of the environments in which these problems/tensions occur. Through the use of observational methods, rich descriptive data can be gathered through which these relationships can be captured.

I also discussed how the choice of how many and which case studies to study is a difficult issue. It is mainly dependant on time and resource constraints, as well as knowing which case studies would be more relevant in answering the research question. These issues will hopefully be resolved in the early part of the second year of study.

The aims, organisational structure and evolution of TransProject was presented. TransProject is the first case study. It is the development of the IT infrastructure of a translational research project at a city hospital in the U.K. The project has been running for 18 months and observational work has been carried out for the last 10 months. I outlined the background, organisation structure and evolution of the project.

Lastly, I outlined the reasoning behind choosing a mixed methods strategy. This is because it is hoped that the range of case studies chosen would include projects that have finished and currently in progress. As a result, it would not be applicable to use observational methods on them. However, with projects that are on-going, first hand observational data could capture richer data than interview techniques. As a result, a mixed methods strategy was chosen in order to capitalise on capturing as much data across the different types and stages of e-science projects as there is. I also discussed the use of observational methods in the first case study.

From the research methodology, the next chapter presents some initial themes emerging from the data collected in the first case study. The data is mainly in the form of transcripts and fieldnotes taken from TransProject's weekly meetings.

Chapter 4: Emerging Themes

In this chapter, I outline some of the emerging themes that have arisen out of the fieldwork carried out in the first case study which was undertaken from the period between March 2005 and November 2005. A number of themes have emerged over the period of fieldwork. Initial data analysis was carried out during the November 2005. The data mainly takes the form of transcripts from weekly meetings or fieldnotes taken around the office.

4.1. “What will the system do?” -The problem of Aligning Objectives

Requirements capture is an important step in the process of systems development. It defines the operations of the system and under what circumstances those operations should be satisfied. It is through this that the developers need to understand and know what to build. It is this clarification and explicitness which helps actors (including the developers and users) understand and have commonalities and a shared idea of what the system should be or could be like. The requirements capture document is often seen as “an agreement between different actors regarding the scope of the project” (Ronkko, Dittrich and Randall, 2005). Moves towards a common understanding of the what the system is and should do.

In the TransProject project, for various reasons, a requirements capture exercise was not carried out. As a result, there is no specific requirements document which is available to everyone on the development team. This has caused a large number of problems – and despite the team being relatively small (4 people – 3 full time and 1 half time), the requirements remains contested. The tension arises from the role of the CancerStudy project upon TransProject. CancerStudy is a project ran by the Lead Principal Investigator (Michael) prior to the TransProject project. The project is coming to the end of its data collection phase (late 2005) and it only collects data

from patients with colonrectal cancer and is nation-wide (cf. TransProject, which is based at one hospital and have collects data for several different cancers). The tension arises because there are differing views from various members amongst CancerStudy and TransProject about what the “new” TransProject system is and will do. There are members of the team who think that the starting point for TransProject would be to replicate CancerStudy, whereas other members of the team think that CancerStudy could be used as a resource, but not as a replica. In this fieldnotes extract, Jim (E-scientist/Developer) and Bob (Research Manager) are discussing what Jim’s work. Jim has just said that he’s been working with the National Minimal Dataset (NMD) for colonrectal cancer, but after the NMD, he needs more information about what other data needs to be stored in the database.

<Data Extract 1>

15th July 2005, Fieldnotes

Jim says, ... The other thing is, is that tables aren’t stable and we’re not sure what people are going to do what and where. How are we going to operate? Has to come from Michael and CancerStudy. Are we going to replicate the CancerStudy [input] screens [on the system]?

Bob says, there’s no reason why not? Unless there’s a good reason not to do that. Its convenient for people who are used to CancerStudy and for the new users, they have to learn the system anyway. Unless there’s a reason not to.

Jim pauses. And says, can do to an extent. I know I’m not producing visible at the moment. But trust me that its going to be there. Replicating CancerStudy is not a big deal. It’s the differences that are interesting for me.

In this conversation, Jim is asking Bob to tell him what other data to collect and the working practices of the system in which he has to build. He explicitly asks Bob whether he is expected to replicate the CancerStudy user interface (the “screens”). Bob’s reply implies yes. Then Jim replies that there should be something more to the TransProject study than just to replicate the CancerStudy process. Jim’s action implies his interests lie in the different addition to CancerStudy when he says “it’s the

differences that are interesting for me”. Here, it demonstrates how there are two differing viewpoints about the TransProject system – that Bob’s view is that TransProject should replicate CancerStudy’ user interface (which implicitly means replicate a substantial part of their working practices). However, Jim wants to focus on the differences between the two studies. Although there is only one instance presented here, the debate between the two views is a recurring source of tension throughout months of development.

Another point to highlight is that some changes in certain specifications will cause a potentially larger impact than others. As Button and Sharrock (1998) points out – decisions are built upon each other. And so even early on a seemingly small re-specification in the problem domain can result in large scale change in the IT infrastructure (especially the database). This is because there are certain things that are easier to change than others later on – for example, it would be relatively easy to add another field in a table, whereas it would be very difficult to migrate to a different database structure. And whilst it is easier to assess the amount of work that needs to be done once those changes have been identified, the identification of the changes only emerges over time.

In the case study, one prominent example of this is a recent change in the database structure of the system. Previously, the team had worked under the assumption that each patient would be recruited once onto the study and for that patient, they would collect a certain amount of data. However, as Jim says in this extract, after a conversation with one of the other Principal Investigators, it is possible for patients to have multiple episodes. Multiple episodes means the recurrence of cancer within the same patient. There are a number of different classifications for episodes, which includes primary and secondary (for patients that have a number of cancers concurrently e.g. lung cancer and mouth cancer) and there are also first and second episodes which indicate temporally the number of times the patient has had cancer (for patients that have recurring cancers).

<Data Extract 2>

8th Nov 2005, TransProject Weekly Meeting, Transcript

Jim: ... We discussed the notion of episodes, you know, because Andrew said, you know, people who have treatment again are very interesting, we need to know about them, erm, I mean we know about them potentially through the cancer registry of course. But, you know, that seems to be that we might want to get blood from them potentially get them to fill out a food frequency questionnaire

Jim explains that he got the information he spoke to Andrew (a Breast Cancer specialist) that there are multiple episodes. He makes the case that although they might eventually hear about recurring episodes from a secondary resource (cancer registry), that they might want to gain additional data or tissue sample upon the recurrence of cancer. By introducing the notion of episodes (i.e. that patients can have more than one cancer), it radically changes the shape of the database structure. Instead of having a one-to-one relationship with person and an episode, there is a one-to-many relationship instead. It can be argued that beforehand, the system was cancer based (initially, the system would have a menu screen with listing the different types of cancer, and being able to view all of the patients with that cancer) whereas with the change, the classification within the system becomes a lot more patient centred (each patient is unique, and there are multiple episodes attached to each patient). This is a change that will effect everything within the system – from the database structure to the consent process, raising questions such as “If a blood sample is required from a patient in a second episode (which could be years after the first), will the original consent from the first episode be valid, or will consent have to be sought after?”

<Data Extract 3>

8th Nov 2005

Internal document presented at TransProject Weekly meeting

Please note that the structure as presented here is different than the one currently implemented in the system. The current implementation assumes a one-to-one relationship between persons and most other datasets and, crucially, does not include the notion of an episode, i.e. it was assuming that person's involvement with TransProject would be a once-through. This was spotted as a problem during a discussion with Andrew who pointed out that cases with two primary cancers would be very interesting and therefore the database

structure needed to enable us to query for such cases. The proposed structure supports this.

What is also interesting is that this was not a point raised previously, chiefly for two reasons. The first is that this was brought up by Andrew – a breast cancer specialist. With colorectal cancer, which Michael specialises in, generally has a lower survival rate and a higher age of onset. Which means that the likelihood of recurrence in Breast cancer is higher – and therefore, it would be logical for the Breast Cancer specialist to point this out. The second point leads from the first, and creates a radical point which will split CancerStudy and TransProject apart. Part of the reason this issue/specification was not discovered earlier is that because of their willingness to copy CancerStudy. When presenting this document, Jim stated that this move would be progression away from how the CancerStudy study would be run:

<Data Extract 4>

8th Nov 2005, TransProject Weekly Meeting, Transcript

Jim: ... Basically, I think it boils down to one key question. CancerStudy had had an exclusion criteria that anyone in the study is not going to be approached again

...

Jim: So there is no such criteria in the TransProject ethics application?

Bob: there shouldn't be

Jim: ok, so if that's the case then, you know,

Bob: there isn't anything written like that at the moment anyway, so I think we'll keep it that way

Jim: so that means we're keeping this notion of an episode and we're going to go and make the required changes to the system, erm ... this then, means there is a significant difference between TransProject and CancerStudy in terms of how they have been run and the kinds of data and the relationships between them ...

At first, Jim clarifies with Bob that they are switching over to being a more episode centred system, and then he makes very explicit that this will differentiate them from the CancerStudy study and that there will be significant differences from now onwards – implying that simply “copying CancerStudy” would not be possible anymore. It is now at this point in which we will see how TransProject will move forward from this.

In this first theme, the lack of clarity in through the lack of a requirements document (or the explicitness of “what the system will do”) means that there is difficulty in finding what the solution in satisfying that problem to be. On numerous occasions, the developers expressed concerns over the lack of requirements. Since the problem is not defined, the solution is equally undefined and not clear. Whilst the e-science vision supports and encourages small scale, multi-disciplinary teams, the problem is not the coordination of such efforts, but rather the coordination of such efforts towards a common goal and the problem of aligning each other’s objectives.

4.2. Roles and Responsibilities

As the lack of clear problem definition and requirements causes tension on the TransProject project, the problem shifts from “What does the system do?” but rather, “Who’s responsibility is it to collect requirements?” In TransProject, matters are rather complicated regarding this. The situation is that the “owners” of the system are the Principal Investigators who manage the project and yet conversely, they are also the **sole** end users of the data processed in the system. It is the nurses that input data into the system, and the developing team that maintain it, but yet, the only end user of the system for the foreseeable future are the P.I.s themselves (although there are plans to extend the system in the future).

In the case study, there are clearly two different viewpoints. Jim, on one hand thinks the “blame” for the lack of direction/systems requirements arises at the P.I. level. As a result, he believes that it is the P.I.s’ responsibility to meet together in one room and have a workshop in order to provide the team with some concrete answers. On the

other hand, Bob thinks that this approach would be futile as it would be very difficult to get the P.I.s in one room for a lengthy amount of time – such as one required for a workshop. Instead, the effort should be made towards getting a “working” system, and then comments (and aggregated into requirements) can be collected from the P.I.s. In this fieldnote extract, both Jim and Bob puts forward their case. It is taken from a weekly meeting relatively early on in the system, Jim starts off by saying to Bob that the P.I.s should have a meeting:

<Data Extract 5>

21st June 2005

Fieldnotes from TransProject Weekly Meeting

Jim asks, so whats next? For me, that’s the 2 most important things. Thinking about it, Andrew asked about the dataset - and how they came to be. The PIs should have a meeting about this.

[...]

Then Bob asks what the purpose of the meeting would be:

[...]

Bob asks, is it just for the risk assessment?

Jim says, no, its for the whole dataset. For them to raise any questions (such as risk assessment). Is Andrew going to carry on or is it going to be done by CancerStudy? They really need to have these discussions. The thing is whether this is done through a couple of phone calls or whether they need to have a meeting about this.

Bob says, I’ll ask Michael about it

Jim says, we’ll need a focus workshop to discuss these things

[...]

Jim establishes that the meeting would be for the P.I.s (principally Andrew and Michael) to get a final confirmation for the dataset i.e. a final confirmation that there will be no additions to the dataset. Note how Jim states that “they really need to have these discussions” emphasising the onus on the P.I.s to talk. Bob replies that this would be difficult due to the P.I.s time commitments:

[...]

Bob says, its all very difficult with this time of year. Its very difficult to get all of them in a room. Someone is bound to be away. We can flag this, but we can't force them in a room together. Are these the fields that are in there?

Jim says, no, its more than that, it's the [working] practices and how the data came to be. There needs to be some agreement. Workshop.

Karen says, but in the last one, no one came.

Bob said that it would be difficult due to the P.I.'s time commitments, and instead, show them a “half-working” system and get them to comment. Jim then insists on the workshop, and Karen (the database manager) reminds Jim that the last time they held a workshop, no one attended.

Another way this tension can be view as is whether it is up to the end user (i.e. the clinicians/scientists) to define these problems, and assign the blame to them when the system does not live up to expectations, or, is it the responsibilities of the developers to coax details and requirements from the end users, no matter how difficult it seems? Bob is using an underspecified system to leverage time with the end users by giving them a half-working system, then asking for comments. Responsibilities for requirements capture then falls upon the developers. In this sense, it is a matter of the developers managing the P.I.s (their managers) as well as the P.I.s managing the developers. Note also, that this problem arises because of the inability of the developers to “demand” the user's time. It can be argued that since the users in this case (the P.I.s) have an unusually large proportion of “power” in this case – i.e. the developing team cannot demand/force the users to devote time to them.

In summary, this theme presented the question “**who’s responsibility** is it to collect the right requirements?” In the case study, there is tension as there are differing views in where that responsibility lies, one developer seeing the onus resting on the P.I.s – who are also the users, and the other seeing the onus is on the developers to constantly enquiring and slowly teasing requirements from the P.I.s. Ultimately, this discussion of responsibility falls onto accountability – who can be called into account should the system not meet expectations? That is the next step of analysis and data collection.

4.3. Data Consistency and Integrity in E-Science

Under the vision of e-science, one of the promises e-science makes is the idea of flexible data. Flexibility in this sense is being able to have scientific data available anywhere, anytime to any scientist, and also the ability re-purpose specific data as well. This vision fits in well with the aims of translational research, which attempts to quicken the process from “basic” bench science to treatment for patients and back. Both are based on flexibility and the ability to move research areas quickly.

A major tension arises when the work of e-science has to work under the rubric of traditional science. Despite the vision of quick and flexible data, the fieldwork suggests that there is high importance placed on continuity and consistency in the data collected and the way that data is collected/analysed. There are two reasons for this emphasis on consistency. The first is that in traditional science, there is a certain amount of consistency needed in order to compare the scientific data. For example, a basic view of how science is conducted is that an experiment is ran according to pre-set parameters. For the next experiment, one would change a pre-defined parameter and run the experiment again, but under the same conditions as the first. Because everything is the same, the changes in the result of the experiment would be collated as a direct result from the change in parameter. The important point here is that in order to compare results, the data must be collected and processed in a consistent manner. And only data that has been collected in the same way can be compared. This, of course, is a problem that was not present before, since each scientist had

responsibility for their own experiments, and each scientist knows what the contingencies and caveats that their data contains. Data from one experiment will be produced under local conditions, with an understanding from the scientist in relation to the time and space in which that data is produced. This is both evident in “traditional” scientific studies (such as Garfinkel’s study of astronomers) as well as in more recent clinical studies. Garfinkel et al (1981 quoted from Button and Sharrock, 1994), for example, discuss how scientific discoveries are “built up”. In their study, Garfinkel et al analysed the recording of two scientists during a night in which they discovered a pulsar. They had to repeat their readings in order to verify the “discovery”. This would only be possible because they understood the fallacies and contingencies upon which the data their equipment produced contained. What is evident in the TransProject project, is that right from the start, they have to define a particular dataset as any data that is collected after that would render it useless unless the data is flagged as such. This way, only like for like data would be compared.

The second reason for this emphasis on consistency is the ethics application (to gain ethics approval) that has to be submitted requires details of the standard operating procedures (S.O.P.s), all documents used (especially the ones used with patient contact) and details of what data would be collected. Without ethics approval, patients cannot be recruited onto the study. Also, following on from clarifying the S.O.P.s, documents and working practices provides the resources from which the requirements can be drawn from. As discussed above, this would greatly clarify the systems requirements and relieve some of the tension that exists within the project.

To ground some of these concepts, I present a data extract from a recent weekly meeting. In the extract, Bob is not present as he is having a meeting with Michael. The rest of the team (Laura, Jim and Karen) are discussing recruitment of patients under the CancerStudy ethics application. Through this, the newly recruited nurses can start training to recruit patients right away. As a result, the team can get some “real” data.

<Data Extract 6>

8th Nov 2005, Transcript of TransProject Weekly Meeting

Jim: ... and I think it's a crucial one, the terms and conditions of consent is different. So speaking in terms of the people recruited once TransProject has ethics approval that's not a problem because for our purposes they will have TransProject consent. We can do with them whatever TransProject can do, erm, in the meantime, until we have consent, we can only recruit them on CancerStudy, and that means that ethics application applies and the things they have on their consent forms applies, which are going to be different

Laura: Hmmm ... now that's a problem actually to have people with different consent that in the database

Jim: exactly

Laura: Because it might be quite clear now, but in the future it might not be. Hm. Yes.

[...]

Jim starts by explaining that patients recruited under the CancerStudy ethics application mean that has different consent from the rest of the TransProject data within the database (i.e. the data collected once the TransProject ethics approval has been gained). Laura agrees and adds that the data might be lost with all the other TransProject data in the future, and hence there will be particular data which should be treated differently, but end up not being treated differently. Jim carries on to explain why this is the case:

[...]

Jim: and I don't want. Cos we, we, we set out to build TransProject as a common infrastructure. We set out to, you know, stop this silly business of having, you know, so many studies with different things so that's why we're going for this generic consent. And I don't want anyone in the database that that's not TransProject consented. Whatever consent they have on top of that when they're taking part in a trial, that's fine but the basic consent should be the same. And I erm, and I would be able to erm, advance human knowledge, but I don't think its worth for this small number of patients, erm,

Laura: to include them?

Jim: yes, it would make things more complicated

Karen: So basically we'd consent the patient to consent to any study in TransProject

Jim: yes, you would have a baseline TransProject consent and then if they are taking part in a trial or if they are in addition to what they normal doing if they are given the pre-operation blood which might not be part of the normal TransProject procedures I don't know yet, you know, there will be additional things that they will consent to and we need to record them. But the baseline TransProject consent about, you know, access to their blood, tissue erm, access to their medical records erm, all of that is going to be a one off thing and they either sign it or they don't. There's no ... they're going to cut down on the options that you can have

[...]

Jim's action implies that the raison d'être of the TransProject system is to provide a common platform, and to have data that is supposed to be treated differently from the start would be additional work that could not be justified considering the number of cases would be less than 15. Also, with the building of a common platform, Jim's view is that **all** data would have some commonality/property. However, this does mean any data they collect prior to ethics approval are rendered useless. Jim then suggests that this data could be used for testing purposes:

[...]

Laura: But one issue now, if we only want to include people only with TransProject consent in the TransProject database that does mean that we can't recruit straight away because we do not have ethics

Jim: well, what my suggestion is that what we can do erm, is we can let the nurses recruit for CancerStudy they will make a copy of

their work and send it off to Maureen erm, and we will be able to use this paperwork to ... erm ... as test data basically. We might as well get our processes right

Laura: Yeah ...

Jim: erm, we will eventually start recruiting TransProject patient in earnest, we can throw away that data

Laura: ok

<End of Data Extract>

This last section sees Laura agreeing that the data should be used for testing, and both Jim and Laura agree that the data only has a shelf life until the TransProject ethics approval is gained (Laura simply refers to it as “ethics”). The ethics approval has to be gained from an ethics committee so that the study can recruit patients onto the study. To gain approval, TransProject must outline how they are recruiting the patients, what documents they will be using, and what they are doing with the data afterwards (i.e. secure storage, purpose of use). Once the approval has been gained, the system would be emptied of the CancerStudy data. However, this does raise the question of whether using the CancerStudy ethics at the start means the TransProject system would be further shaped by CancerStudy.

4.4. Summary

This chapter presented three emerging themes from the data collected in the first case study – TransProject. The themes are the problem of aligning objectives; roles and responsibilities and rigidities in datasets in e-science projects.

The first of these themes discusses the work of aligning objectives within an e-science project. I raised two important points here. The first being the work of trying to understand “what the system will do?” or “what the system should do?”. Since the details of how the system should work has not been specified, the developing team have found it difficult to understand what they are trying to build and in doing so,

they have disagreed on the priorities of what should be built next. The second point was how changes in certain specifications result in large scale changes to the system. The data presented revolved around how the team had initially built the system, then when one of the specifications change – i.e. the need to store multiple recurrences of cancer – meant a significant shift in the database structure. This change shifted the application from being cancer centred towards being patient centred.

The second theme presented revolves around the question “who is responsible for producing good requirements?” The issue is that the P.I.s (who are responsible for the overall progress of the system) are also the end users of the system. The issue is that as end users, it is where “traditionally” the developers would gather requirements from. However, due to the P.I.’s time constraints it presents a very difficult challenge for the developers to gather requirements from them, or even, just to specify requirements from scratch. It then presents us with an interesting look onto the views the roles of each member of the development team, and also the responsibilities of the project.

The third theme revolves around the flexibility of data as hailed by the vision of e-science, and the data consistency and integrity that has to be imposed through: consistency to compare data; the regulatory structure of “traditional” medical research (in order to gain ethics approval) and the need for basic specifications for the system. This theme examines the tensions in “doing” e-science in the current environment of the scientific community.

All these themes highlight interesting dynamics that occur during the development of e-science systems. It is hoped that over the course of the next two years, these themes would be further explored and expanded within TransProject and also across the other case studies. Leading on from the emerging themes, the next chapter discusses the work done already during the first year of study, and will outline the work proposed (which includes further exploration of the above themes) in the second and third year of study.

Chapter 5: First Year Progress and Proposed Work Plan

5.1. First Year Progress

During the first year, progress was made on three fronts. The first was in terms of background reading on e-science systems, ethnography/observational research methods, and systems development literature – especially in relation to empirical work on software engineers; most of which is distilled in this report. The second was in terms of attending various workshops and conferences throughout the year - some of which involved giving a talk or preparing materials. Most notable was an abstract paper in the Doctoral Consortium of the BCS Human Computer Interaction Conference. The third was fieldwork carried out at the first case study.

Ten months of ethnographic work has been carried out since Jan 2005. The work has mainly been through observations in team meetings/developer's conversations at TransProject. Formal meetings started being held in July 2005, and I have made notes and also agreed to write the minutes of all the weekly team meetings. Some meetings were recorded with permission, with on average, once per month.

In addition to weekly team meetings, I spent time around the TransProject office observing. On average, this would be 1 or 2 days per week. Observations would be made on what the developers were doing, what they discussed, how they reacted to certain issues and what concerns they had. Ethnographic notes were taken during the meetings and observations.

I also attended some meetings which took place between team members and outside organisations. These were only meetings held with one particular organisation, a previous medical study in which the lead Principal Investigator led. These meetings

tended to outline the working practices (both formally defined and “informal” ones) and it acted as a way of transferring information from one site to another.

5.2. Proposed Work for second and third year

For the remaining two years, the proposed work is that I continue with the empirical work. With the first case study, I would hope to carry out the ethnographic study for a further 12 months. Meanwhile, selection of the second and third case studies has to be made and access negotiated. Note that this is a tentative plan, and as with observational studies exact point of departure is difficult to predict due to its exploratory nature.

I propose that data be continually collected throughout the coming months, and depending on how much/little is being gained from carrying on with the case studies, the period of study will vary. As the same with the first ten months of work, the data collected will consist of observations and semi-structured interviews resulting in audio recordings and more importantly, fieldnotes.

I am aiming to do collate and do some data analysis during the Christmas period of 2005, as work at the TransProject office will be slower (and hence, less need to be around). This will provide me with an opportunity to work on data analysis without the pressures of being in the TransProject office writing fieldnotes (a time consuming task), nor having to write the minutes of the meetings. Both of which are highly time consuming activities. In addition, as Agar (1980) notes, a certain amount of “distance” needs to be achieved in order to analyse ethnographic data.

By Nov 2006 (or by the start of Jan 2007 at the latest), I aim to finish fieldwork and spend the remainder period (10-12 months) analysing the data collected, alongside the write up of thesis. This is aimed as a hard deadline since a period of time will be required to write the thesis.

Chapter 6: Summary

This research proposal has outlined the background, research methodology, emerging themes and proposed work of the thesis. I started by describing the context and the need to study e-science systems. E-science has been hailed as a new way of doing science which involves more collaboration, more computation power and a flexible service orientated architecture. With the additional collaborations and flexibility, it increases the complexity of traditional systems which support the work of “science”. Their novelty presents interesting socio-technical challenges. Organisational, political, social and technical factors play a significant part in the shaping of such systems. In this proposal, I have outlined the reasons why requirements capture should be seen as a socio-technical process. With the new complexity that e-science brings, I want to examine the complex tensions and interplays that occur within requirements capture in the development of an e-science project.

I have outlined the literature review of the thesis. The two main body of literature are to do with the vision of e-science and requirements capture. The vision of e-science is one which encourages big science to be done on a global collaborative distributed level. I outline the challenges such a vision faces, such as IPR, sharing of data and development with bleeding edge technologies. I presented reasons why the requirements capture process is a socio-technical process rather than a purely technical one. The process is often long and complex, as it is a learning/knowledge transfer process through an understanding of the issue and/or changes to the external environment. It is this conceptualisation of the requirements process as a knowledge transfer process that I think will be useful in the study of requirements capture in e-science systems.

The research approach that I have decided to conduct is a qualitative study, using three case studies with a mixed methods approach to data collection. It is one which attempts to capture the complex relationships that occur during systems development. The difficulty is in choosing two further case studies as e-science projects; as projects

vary in a number of ways – some of which are more significant than others. This is a matter to be resolved during the second year of study. In addition, access and practicalities (such as time and resources) have to be resolved in conducting these two case studies.

I have presented three emerging themes from the initial period of study; the problem of aligning objectives in requirements capture, roles and responsibilities for good requirements and finally, the tension of rigidities and flexibilities in e-science. The first theme discusses the work of aligning objectives amongst team members in an e-science project. This is magnified with the multi-disciplinary teams that are often used in the development of e-science systems. The second theme discusses the notion of roles and responsibilities that occur in e-science. It especially highlights the special relationship which occurs in the case study, namely that the end users of the system are also responsible for the overall direction of the project. Due to their busy schedules, the developers find it very difficult to ask for time and it is this managing of P.I.s that the developers have to contend with. The third theme discusses the tension between rigidities and flexibilities as one of the key challenges of e-science. The vision of e-science is to create flexibility and the ability to move quickly – a vision shared within translational research (which is the objective in the case study). The tension lies between this vision, and the regulatory body (to gain ethics approval in order to deal with patients) and the developers (to have a basic understanding of what needs to be built and how). There is a need to define specific characteristics of the system (e.g. datasets, working practices), which, once decided, incurs a high cost to change. Yet at the same time, there is a need build flexibilities into a system which will allow the vision to be realised. It is how various members of the development team deal with this tension that is of interest.

Finally, I presented on progress in my first year, and set out an outline of what I am planning to do for second and third year. First year progress was investigating literature on ethnomethodology, requirements capture and systems development literature. I also did 9 months of observational work at the first case study, as well as attending various workshops and providing parts of talks. The proposed work for second and third year is to gain access to a further two case studies, both with varying characteristics. Data from the studies will be collected and analysed throughout

second and start of third year, with the majority of third year being allocated to writing the thesis.

Bibliography

Anderson, R. J., Hughes, J. A., Sharrock, W. W. (1987) "Executive Problem Finding: Some Material and Initial Observations" *Social Psychology Quarterly* 50(2): 143-159

Agar, M. (1980) *The Professional Stranger: an informal introduction to ethnography*, Academic Press: New York

Akrich, M. (1992). "The De-Description of Technical Objects", in Bijker, W and Law, J. (eds) *Shaping Technology/Building Society: Studies in Sociotechnical Change*, MIT Press: Cambridge and London

Asaro, P.M. (2001) "Transforming society by transforming technology: the science and politics of participatory design" *Accounting Management and Information Technology* 10: 257-290

Atkinson, M., Crowcroft, J., Goble, C., Gurd, J., Rodden, T., Shadbolt, N., Sloman, M., Sommerville, I., Storey, T. (2002) "Computer Challenges to emerge from eScience", document accessed from <http://www.semanticgrid.org/docs/Vision.pdf> on 6th Dec 2005

Bergman, M., Leslie King, J. and Lyytinen, K. (2002) "Large Scale Requirements Analysis as Heterogeneous Engineering" *Scandinavian Journal of Information Systems* 14: 37-55

Brooks, R. (1990) "Categories of Programming knowledge and their application" *International Journal of Man-Machine Studies* 33(3): 241-246

Button, G. and Sharrock, W. (1994) "Occasioned Practices in the Work of Software Engineers" in *Requirements Engineering*, Jirotko, M. and Goguen, J.A. (eds) Academic Press: New York

Button, G. and Sharrock, W. (1996) "Project Work: The Organisation of Collaborative Design and Development in Software Engineering" *Journal of Computer Supported Cooperative Work* 5: 369-386

Button, G. and Sharrock, W. (1998) "The Organizational Accountability of Technological Work" *Social Studies of Science* 28(1): 73-102

Curtis, B., Krasner, H., Iscoe, N. (1988) "A Field Study of the Software Design Process for Large Systems" *Communications of the ACM* 31(11): 1268-1287

DeRoure, D., Jennings, N., Shadbolt, N. (2001) "Research Agenda for the Semantic Grid: A Future E-Science Infrastructure" accessed from http://www.nesc.ac.uk/technical_papers/DavidDeRoure.etal.SemanticGrid.pdf on 6th Dec 2005

Dourish, P. and Button, G. (1998) "Technomethodology: Foundational relationships between ethnomethodology and system design" *Human-Computer Interaction* 13(4): 395 - 432

Emerson, R. M., Fretz, R. I. and Shaw, L. L. (1995) *Writing Ethnographic Fieldnotes*, University of Chicago Press: Chicago and London

EPSRC (2004) "e-Science 2004: The Working Grid", EPSRC Report on E-Science

Fetterman, D. (1989) *Ethnography Step By Step*, Sage: London

Fox, G. and Walker, D. (2003) "e-Science Gap Analysis" accessed from http://www.nesc.ac.uk/technical_papers/UKeS-2003-01/GapAnalysis30June03.pdf on 6th Dec 2005

Garfinkel, H., Lynch, M., Livingstone, E. (1981) "The Work of a Discovering Science Construed with Material from the Optically Discovered Pulsar" *Philosophy of the Social Science* 11: 131-158

Guidon, R. (1990) "Knowledge exploited by experts during software system design" *International Journal of Man-Machine Studies* 33(3): 279-304

Hartswood, M., Procter, R., Slack, R., Voss, A., Buscher, M., Rouncefield, M., Rouchy, P. (2002) "Co-realisation: Towards a principled synthesis of ethnomethodology and participatory design" *Scandinavian Journal of Information Systems* 14(2): 9-30

Hartswood, M., Jirotko, M., Procter, R., Slack, R., Voss, A. and Lloyd, S. (2005) "Working IT out in e-Science: Experiences of requirements capture in a HealthGrid project", *Proceedings of HealthGrid 2005*, Solomonides, T and McClatchey, R. (eds) IOS Press: Amsterdam, Berlin, Oxford, Tokyo, Washington

Hertzum, M. (2004) "Small-Scale Classification Schemes: A Field Study of Requirements Engineering" *Journal of Computer Supported Cooperative Work* 13: 35-61

Hinds, C., Jirotko, M., Rahman, M., D'Agostino, G., Meyer, C., Piper, T and Vaver, D. (2005) "Ownership of Intellectual Property Rights in Medical Data in Collaborative Computing Environments" in *Proceedings of the First International Conference on e-Social Science*, Manchester

Jirotko, M., Procter, R., Hartswood, M., Slack, R., Simpson, A., Coopmans, C., Hinds, C. and Voss, A (2005) "Collaboration and Trust in Healthcare Innovation: the eDiaMoND Case Study" *Journal of Computer Supported Cooperative Work* 14: 369-398

Law, J. (1987) "Technology and Heterogeneous Engineering: The Case of Portuguese Expansion" in *The Social Construction of Technological Systems*, Bijker, W., Hughes, T. and Pinch, T (eds) MIT Press: Cambridge, MA

MacKenize, D. and Wacjman, J (eds) (1985) *The Social Shaping of Technology*, Open University Press: Milton Keynes and Philadelphia

Momtahan, L. and Martin, A. (2002) "e-Science Experiences: Software Engineering Practice and the EU DataGrid" in Proceedings of APSEC2002, Queensland

Monteiro, E. and Svanæs, D. (1993) "The role of empirical evidence in software engineering", Norwegian Informatics Conference, Oslo

Mumford, E. (1987) "Sociotechnical Systems Design: Evolving Theory and Practice" in Computers and Democracy, Bjerknes, G., Ehn, P., and Kyng M. (eds) Avebury: Aldershot

Ronkko, K., Dittrich, Y., Randall, D. (2005) "When Plans do not Work Out: How Plans are Used in Software Development Projects" Journal of Computer Supported Cooperative Work

Sommerville, I (2001) Software Engineering, 6th edition, Addison Wesley

Walz, D.B., Elam, J.J., Curtis, B. (1993) "Inside a Software Design Team: Knowledge Acquisition, Sharing, and Integration" Communications of the ACM 36(10): 63-77

Winner, L. (1985) "Do Artefacts have Politics" in The Social Shaping of Technology, MacKenize, D. and Wacjman, J. (eds) Open University Press: Milton Keynes and Philadelphia

Woolgar, S. (1996) "Technologies as Cultural Artefacts" in Information Communication Technologies: Visions and Realities, Dutton, W (ed) Oxford University Press: Oxford