View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by UCL Discovery

Stakeholder Involvement, Motivation, Responsibility, Communication: How to Design Usable Security in e-Science

Ivan Flechais
Oxford University
Computing Laboratory
Wolfson Building
UK – Oxford OX1 3QD
ivan.flechais@comlab.ox.ac.uk

M. Angela Sasse
Department of
Computer Science
University College London
UK – London WC1E 6BT
a.sasse@cs.ucl.ac.uk

e-Science projects face a difficult challenge in providing access to valuable computational resources, data and software to large communities of distributed users. On the one hand, the raison d'être of the projects is to encourage members of their research communities to use the resources provided. On the other hand, the threats to these resources from online attacks require robust and effective security to mitigate the risks faced. This raises two issues: ensuring that (1) the security mechanisms put in place are usable by the different users of the system, and (2) the security of the overall system satisfies the security needs of all its different stakeholders. A failure to address either of these issues can seriously jeopardise the success of e-Science projects.

The aim of this paper is to firstly provide a detailed understanding of how these challenges can present themselves in practice in the development of e-Science applications. Secondly, this paper examines the steps that projects can undertake to ensure that security requirements are correctly identified, and security measures are usable by the intended research community. The research presented in this paper is based on four case studies of e-Science projects. Security design traditionally uses expert analysis of risks to the technology and deploys appropriate countermeasures to deal with them. However, these case studies highlight the importance of involving all stakeholders in the process of identifying security needs and designing secure and usable systems.

For each case study, transcripts of the security analysis and design sessions were analysed to gain insight into the issues and factors that surround the design of usable security. The analysis concludes with a model explaining the relationships between the most important factors identified. This includes a detailed examination of the roles of responsibility, motivation and communication of stakeholders in the ongoing process of designing usable secure socio-technical systems such as e-Science.

Keywords: Security, security design, e-science, socio-technical design, usable security

1 Introduction

e-Science projects aim to provide useful scientific services (such as computational resources or easy access to large amounts of data) in a distributed environment. One of the biggest challenges e-Science projects face is to make valuable assets (high-end

computing power, storage capacity and bandwidth, and the data and software they contain) available to a large - potentially globally - distributed community of legitimate users, and at the same time protect those assets from unauthorised use and undesirable consequences.
The challenge of designing security that is effective but usable is not unique to e-Science. Research into usable security over the past 10 years (e.g. [Adams & Sasse, 1999; Whitten & Tygar, 1999]) has demonstrated that users actively avoid security mechanisms that are difficult to use, and/or make mistakes that undermine security. Resolving this challenge is particularly relevant for e-Science which aims to bring different communities of users together, whilst maintaining a secure environment.

There is thus a need in e-Science for both usability and security for the following reasons:

1. To attract and retain users from the target research communities, the system needs to offer resources that are perceived as valuable, however these resources are also of value to a range of attackers.
2. The security needs of stakeholders contributing or sharing resources must be met, or else they will not provide these resources.
3. At the same time, the system must be usable and accessible to the target research community, i.e. the security mechanisms that protect resources must not be so onerous that they deter researchers.

e-Science projects should therefore consider the security needs and capabilities of all its various different stakeholders in a bid to address the issues of both usable and effective security.

The biggest problem in achieving this is a lack of guidance for designing secure and usable software systems. As a means of addressing this we have previously presented AEGIS, a secure socio-technical system design method aiming at achieving usable security in software systems. As part of its ongoing design and validation, AEGIS was empirically applied in four e-Science projects. More details can be found in [Flechais, Mascolo et al, 2006; Flechais, Sasse et al, 2003; Sasse & Flechais, 2005].

The empirical application of AEGIS in security design sessions with e-Science projects provided an opportunity to explore the issues surrounding usable security design in greater detail. Most e-Science projects are expected to operate in a distributed setting, and many are also being developed in a distributed environment, through a consortium of different parties (for example laboratories, universities, and private companies). In practice, this has led to a number of problems which fall in two main categories:

1. Designing and devising secure systems that are usable by widely distributed user groups.
2. Overcoming the difficulties that arise out of having a development team that is geographically spread out.

The research presented in this paper aims to provide knowledge about the factors and issues that surround the practical design of usable security. Based on the analysis of the real-world application of AEGIS to four e-Science projects, we describe the lessons learnt from these case studies. Key findings are the importance of assigning responsibility

of security, making security requirements explicit to motivate all stakeholders to take their share of responsibility, and the importance of clear communication about security between different stakeholders. We analysed the transcripts of the studies in detail and present a model and description of the general factors and issues that were identified during security design. A review of the background literature in usable security design can be found in section 2, followed by a description of the research methodology adopted and studies conducted in section 3. Section 4 presents the analysis of the results, followed by a discussion of their relevance to secure system design in section 5.

2 Background

In an effort to achieve best practice, many e-Science applications have been developed following rigorous software engineering processes. In principle, software engineering provides all the tools and the techniques necessary to build systems that fulfil customer expectations, including security. However in practice, the continuing high number of security incidents in industry [Department of Trade and Industry, 2006], coupled with an increasing amount of research that shows that security is not well-suited to human factors [Adams & Sasse, 1999], indicates that current software engineering practices are not enough.

2.1 Information Systems Security

In the field of information systems security, a small number of security design approaches, called *"integrative approaches"* by [Siponen, 2001] attempt to consider organisational needs. These include responsibility modelling (Structures of Responsibility [Backhouse & Dhillon, 1996], and abuse cases [McDermott & Fox, 1999]), a managerial approach to systems risk [Straub & Welke, 1998] or security modifications to existing information systems development methods [Hitchings, 1996; James, 1996].

Whilst these approaches provide some means of addressing user issues in security, they do not solve completely the problem of developing secure and usable software systems. They either do not integrate seamlessly into the development cycle (forcing developers into tools, models and techniques they may not know or wish to adopt) or fail to address issues of *development duality* – the problem of developing security separately from functionality. In addition to this, *"… explanations come enshrouded in complexity, largely because of the sophisticated sociological and philosophical bases, and as a result the audience for such security approaches remains just a small group of academic researchers."* [Dhillon & Backhouse, 2001]
Finally, little is known about the dynamics or the factors that surround the application of these methods, which could improve current practice and inform future research.

2.2 Human Computer Interaction and Security

The research discipline of Human-Computer Interaction (HCI) is now well-established, with multiple conferences and publications dedicated to the subject. Central to HCI is the concept of human factors and how these affect and shape how people react to computer systems. Human factors typically refer to the intrinsic properties of people, such as short-term memory, visual acuity or physical dexterity but they can also refer to more systemic and organisational issues such as cultural and social effects. These properties can strongly influence the design of a system, most visibly at the interface level, but also at a more fundamental level such as the underlying model of operation of the system.

In the field of computer security, the importance of human factors was first voiced in 1975 when the notion of *psychological acceptability* [Saltzer & Schroeder, 1975] was proposed as a desirable property for computer protection mechanisms. The idea presented in that paper was later taken up by [Zurko & Simon, 1997], who argued for user-centred security design to ensure that users, developers and system administrators made correct decisions. Studies of existing security mechanisms, such as passwords [Adams & Sasse, 1999] or PGP [Whitten & Tygar, 1999], also highlighted the important role that users play in the overall success of secure systems. As a consequence of this growing research, the new field of Human Computer Interaction in Security (HCISec) has become increasingly recognised.

To date, most the research in HCISec has largely focused on users (such as examining passwords, security visualisation or encryption mechanisms). Unfortunately, assisting developers in designing usable security, as identified by Zurko & Simon, has not been the focus of much attention. Neither have the wider systemic issues of designing usable security been explored significantly.

Current HCISec design techniques fall into two categories: design guidelines and usability evaluations of secure systems.

A set of user interaction design guidelines for secure systems was proposed [Ka-Ping, 2002]. Some examples of these guidelines are:

> *"Path of Least Resistance. The most natural way to do any task should also be the most secure way.*
>
> *Appropriate Boundaries. The interface should expose, and the system should enforce, distinctions between objects and between actions along boundaries that matter to the user.*
>
> *Revocability. The interface should allow the user to easily revoke authorities that the user has granted, wherever revocation is possible."*

Whilst these guidelines are useful as a reminder to developers to consider user needs, they do not tell them how to achieve these aims. The *revocability* guideline, for example, specifies that users should be able to easily revoke authorities – which is a very important part of the security of the system – but not how to achieve this. The guideline leaves this as a problem the developer should resolve.

In terms of helping in the design of usable and secure systems, these guidelines only state desirable properties that may (or may not) apply to any particular system.

At present, the only effective means of ensuring that a secure system is usable is to periodically conduct evaluations and test user responses. As can be seen from the PGP usability evaluation [Whitten & Tygar, 1999], this is useful as a means of uncovering problems. One major issue with this is that as a design practice it has no prescriptive value, and therefore does not inform developers about how to achieve a usable and secure design.
An additional problem is that designing, conducting and interpreting an evaluation currently requires specialist knowledge. Whilst in the field of HCI this is common practice, this knowledge is not widespread in the security community and this poses an additional difficulty.

2.3 Developing Usable Secure Systems

A usable secure system design method must reconcile both social and technical aspects of the system with the goal of assuring that desirable interactions are actually carried out and undesirable interactions are prevented, detected, reacted to, deterred or avoided.

AEGIS [Flechais, Sasse et al, 2003] was formulated as a software engineering process to guide the elicitation of requirements, identification of risks to the system and selection of countermeasures. Since the process is designed to fit into a normal software engineering development process, this addresses the issues identified above (section 2.1) such as development duality or unfamiliar tools. In addition to this, AEGIS also aims to achieve usable security through:

1. **Participative design**. This involves including users and other stakeholders in design, and follows from the seminal research of Enid Mumford and her colleagues [Mumford, 1983; Mumford & Weir, 1979]. It has been frequently argued that participation in information system design is very important to the success of a system [Barki & Hartwick, 1989; Mumford, 1983; Mumford & Weir, 1979; Rousseau, 1989; Tait & Vessey, 1988; Wong, 1994]. Studies [Butler & Fitzgerald, 1997; Irestig, Eriksson et al, 2004; Wong, 1994] of participative design practices, such as the ETHICS method (Effective Technical and Human Implementation of Computer-based Systems) [Mumford, 1983], suggest that whilst user involvement does not guarantee a successful system design, the use of a participative approach does foster a climate that is conducive to successful development, and can also lead to more pragmatic designs [Irestig, Eriksson et al, 2004].
   The key element of participative approaches revolves around representing the relevant viewpoints of different parties in such a manner as to achieve a consensus.
2. **Contextual modelling**. Based on the *contextual design* approach [Beyer & Holtzblatt, 1998], AEGIS uses a modelling notation that allows the representation of the system and its users in their environment. The core principle of contextual design is to gain a detailed understanding of the needs and working practices of

customers and other stakeholders in the system. This then supports developers in building systems that integrate well into the working environment of the user.

2.4 Summary

Software engineering approaches to security are generally focussed on providing technical security, but there is growing evidence [Saltzer & Schroeder, 1975] that these approaches do not consider the needs of people sufficiently [Zurko & Simon, 1997]. Although developers have been identified as being a target group that requires usable security [Zurko & Simon, 1997], no efforts seem to have been made to ensure security development methods are well suited to the needs of developers. In fact, by putting the onus on the developers to build better user interfaces, the current trend in HCISec research is arguably adding to the complexity of building secure systems. And whilst *interpretive* security approaches (see section 2.1) have started to address the needs for a socio-technical approach to secure systems, they are still falling short in providing a practical and relatively simple means for developers to achieve this.
Consequently AEGIS was developed and empirically tested as a means of addressing the complex problem of achieving usable security design. As will be seen in the following case studies, this type of approach proved particularly useful in e-Science as a means of identifying the security needs of different stakeholders, and also in designing mechanisms that would be suited to the intended users.

3 Case Studies

In order to gain a deeper understanding of security and usability needs in e-Science, we conducted 4 case studies with e-Science projects. Whilst the primary purpose of the case study was to trial the AEGIS process, the case studies also provided a detailed source of empirical data on the design of e-Science systems, and their security requirements. Details of the methodology employed in conducting the case studies and analysing the results are presented next.

3.1 Methodology

3.1.1 Action Research

Interventionist research approaches, such as *action research*, are well suited to empirical security research [Baskerville, 1999]. For example, they can benefit the participating organisations by actively intervening and improving on specific problems within the organisation. This promise of immediate assistance seems to be much more persuasive in gaining organisational interest than *"… altruistic arguments about how (the) research might benefit software engineers generally"* [Butler, 2000]. As a means of both elaborating and validating a usable secure system design method, action research was adopted as the research methodology.

The intervention methodology (AEGIS) is a participative design method aimed at reconciling security, usability and software engineering good practice. The participation

of stakeholders in the system design, together with the modelling of the users and operating context of the system are key elements of AEGIS for addressing the social aspects of secure system design. The process of AEGIS identifies security needs by modelling assets and attributing values to these according to security properties (such as confidentiality, integrity or availability). A risk analysis is then conducted to identify the most important areas of the system. Finally, security countermeasures are proposed based on these risks, cost and an assessment of their ease of use in the context of operation. More details can be found in [Flechais, Mascolo et al, 2006; Flechais, Sasse et al, 2003; Sasse & Flechais, 2005].

3.1.2 Grounded Theory

In contrast to these previous publications, the research presented in this paper does not focus on the outcomes of the intervention methodology, but rather on the detailed analysis of the transcripts of the design sessions, as a means of learning more general lessons about the design of usable security in e-science projects. This analysis was conducted using *grounded theory*; a theory building analysis methodology which has proven useful in gaining a better understanding of security issues, such as those surrounding users' perceptions of privacy in multimedia communications [Adams & Sasse, 2001].

According to [Martin & Turner, 1986], grounded theory is *"… an inductive, theory discovery methodology that allows the researcher to develop a theoretical account of the general features of a topic while simultaneously grounding the account in empirical observations or data."* Since grounded theory is a theory-building qualitative analysis tool, it is argued to be particularly suitable for areas in which a detailed description or understanding of the phenomenon does not exist.
The main analytical process of grounded theory consists of taking data, breaking it down, conceptualising it and reassembling it into new forms. The end result of the analysis is a theory that explains and is based on the data.

3.1.3 Research Approach

AEGIS was applied to four projects – EGSO, CLEF, BioSimGrid and DCOCE – with a total of eight workshops conducted. Each workshop lasted between two and three hours, was recorded and transcribed in full. These interventions showed AEGIS to be a useful methodology and also provided useful feedback for improving the method (see [Flechais, Mascolo et al, 2006; Flechais, Sasse et al, 2003] for more details about the specific case studies).

An overview of each case study will now be presented together with the most important issues that were uncovered, followed in section 4 by the grounded theory analysis of the transcripts of these studies.

3.2 European Grid for Solar Observations (EGSO)

3.2.1 Overview

The EGSO project is run by a consortium of different global partners (including among others British, French, Italian and American institutions), with a heavy emphasis on academic participation. EGSO is funded under the Information Society Technologies (IST) thematic programme of the European Commission's Fifth Framework Programme. The project is one of many partners from across Europe that co-operate through the EU GRIDSTART initiative. [EGSO, 2005]

The purpose of EGSO is to provide a Grid making the solar observations of a number of different observatories and institutions available to customers, scientists in particular. EGSO is intended to operate as a virtual observatory, providing a platform through which scientists can access solar observation data from around the world. In addition to providing access to solar data, EGSO also intends to provide a distributed computation service for analysing the data.

The case study involved three stakeholders: two developers from EGSO and one manager. This manager was a researcher himself, and thus part of the intended user community of EGSO.

3.2.2 Issues Uncovered

Security Requirements

During the study it became apparent that the project had very competent software engineers, and that a rigorous software engineering approach was applied to EGSO. This could be seen in documented use cases, requirements validation, user interface design and UML system design. The need for security had been acknowledged and some use cases, albeit in vague terms, described the need for some security mechanisms (e.g. the need for *"direct access to satellite data in near real-time, perhaps only with necessary authorisation"*).

When first asked about security, however, the participants stated that *"no one is in charge of security"*. As with many e-Science projects, EGSO was developed by a variety of geographically distributed stakeholders. As a result of this, communication and coordination between different stakeholders is restricted, reducing the chance of opportunistic, informal interactions – a phenomenon that has been dubbed the 'water cooler effect' [Gorlenko, 2005]. In more centralised development environments, this might be a way of raising security issues and eventually bringing them to the fore, but in a distributed development project, discussions are more likely to stick to a pre-defined agenda. One critical means of making sure that security is not overlooked is to ensure a clear assignment of responsibility for the achievement of specific outcomes. This issue will be examined in more detail in sections 4.2 & 4.3.

It was also stated that security had not been considered in depth because the project was *"still in (the) early stages (of) going from requirements to design"*. Another comment justified a lack of concern for security by insisting that functionality was much more important at this time, and that security would be addressed later. In this case, security was considered as a *non-functional requirement* and the decision to address security at a later stage is a good example of *development duality* [Siponen & Baskerville, 2001].

Prior to our intervention, the only security requirement identified in EGSO was confidentiality (the property of security that concerns unauthorised disclosure of information) and access control, while other important requirements such as availability (the property of security that concerns unauthorised withholding of information) and integrity (the property of security that concerns unauthorised modification of information) had been overlooked.

After the participants had been asked to identify key assets and review a number of attack scenarios, security requirements for availability and integrity quickly emerged. For example, the scenarios in which the data that was assumed to be public could be modified to suit a particular attacker, where user software running through a user executable code service could be used to attack the system, or where a third party gained access to a user's personal space. The impact of these particular attack scenarios on stakeholders ranging from the users (scientists) to the data providers (solar observatories) demonstrated the importance of these additional requirements. Indeed it quickly became apparent that confidentiality was only a marginally important requirement when compared to integrity and availability.

Understanding of Security

The design sessions uncovered inaccuracies in the understanding of security technology for example, the notion that middleware would *"take care of the PKI"* (Public Key Infrastructure) in a digital certificate scheme. The underlying assumption in this statement being that a PKI only requires the design of software. But a PKI, as the acronym describes, is a security *infrastructure* that involved users actively being issued with digital certificates and having to manage these. It is thus also necessary for the social elements of the infrastructure to be designed and implemented.

The assumption made was a convenient one, motivated by a desire to avoid having to use digital certificates. The justification was another assumption: that only a few users would need access to sensitive computing services. However, a more detailed inspection of use cases revealed that, although few users needed access to these computational facilities, other aspects of the system (such as personal user spaces) were also at risk and required strong authentication.

The desire to avoid using digital certificates was as a result of their perceived excessive costs, such as poor usability and the need to implement the necessary infrastructure and training.

3.3 Clinical e-Science Framework (CLEF)

3.3.1 Overview

The purpose of CLEF is to provide a framework through which clinical patient information can be accessed for the purposes of medical research. The "*capture, integration, and presentation of descriptive information is a major barrier to achieving such a framework. Clinical histories, radiology and pathology reports, annotations on genomic and image databases, technical literature and Web based resources all typically originate as text. Often they are dictated and then typed; alternatively they are laboriously coded or annotated manually, usually in incompatible formats that lack rigour and hence cannot be scaled up or aggregated effectively.*" [CLEF, 2004] Because of legal and ethical constraints placed on clinical research, one of the main areas of research for CLEF is in the areas of security, and how to preserve the confidentiality of patient information whilst achieving a useful research framework. The main mechanism being developed by CLEF involved the anonymisation of patient information as a means of protecting patient privacy.

This case study involved two stakeholders from the project. Both had extensive experience in the field of medical research and were also security experts responsible for the security in CLEF.

3.3.2 Issues Uncovered

Security Requirements

One area of concern was the need for integrity in the data. The anonymisation process described by CLEF consisted of removing what they called directly identifiable information from the data: names, addresses and dates of birth of patients. The anonymisation process also consisted of coding the medical information held in the patient records into a standard format, further reducing the possible existence of directly identifiable data. This is a good example of a usability/security tradeoff, where the need for privacy of patient data is seen as being important enough to warrant reducing the amount of information available to the researchers.
From the point of view of scientists, however, the question was how much this process of changing the records affected the integrity of that data, bearing in mind that the ultimate purpose of the data was to support clinical research.
Since the project was designed to research issues such as this, the participants were not able to quantify the impact of this process; however they were able to present three reasoned arguments:

1. Clinical research cannot be conducted on data without first coding it in some fashion. Whilst the clinical coding does change the data, it is necessary to support any research.
2. This would only be critical if the data changed to the point where the conclusions drawn from its analysis would be different from those conducted on the identifiable data. It was argued that research methodologies allowed for some

degree of variation in the data, and that therefore the changes to the data would be taken into account.

3. The purpose of CLEF is to provide a means of informing new clinical studies, as opposed to replacing them. Therefore the data provided by CLEF is intended as a means of facilitating the identification and specification of new research.

The security requirements for privacy were not directly derived from patients, but instead imposed by legal precedent and ethics committees. Based on those guidelines, there should be no way to re-identify a patient (backtracking). However, user-centred research on privacy [Adams & Sasse, 2001] established that – unlike regulation – people do not seek to avoid all risks to privacy. Rather, they balance risks against potential benefits. Thus, given appropriate safeguards, patients might want to benefit directly from new research results (such as for example research identifying new risk factors for diseases). However, because of the need for complying with a perceived level of confidentiality, this facility was disregarded.

> *"I mean effectively, what we're going to do, we recognise that backtracking could be valuable in certain cases, in many ways in order to get it adopted we're saying we're not going to have backtracking, we won't have processes that allow backtracking, though technically, it might be possible. But as far as we're concerned we're not going to set up processes that are going to do that."*

Understanding of Security

A significant difference between the two case studies can be seen in the participants' difference of awareness and knowledge of security. Whereas the participants in the EGSO case study were developers and managers, the CLEF participants were security experts.

In addition to identifying requirements that were particularly important to certain stakeholders (such as the usefulness of backtracking for patients, or the importance of research integrity for scientists), questioning the design of the system also had the effect of conducting sanity checks against these requirements.

In one instance the complexity and cost of the proposed system design was examined when rated against the needs of researchers and the privacy requirements of the patients providing the data.

This cost-benefit discussion allowed the participants to seriously examine their design and justify it. This following quote illustrates this:

> *"I thought about that very hard, and I felt that actually although it was very helpful that you prodded me in the direction, I actually thought no I actually think he's wrong and I'm right, and that we are right."*

3.4 Biological Simulation Grid (BioSimGrid)

3.4.1 Overview

"The aim of the BioSimGrid project is to make the results of large-scale computer simulations of biomolecules more accessible to the biological community. Such simulations of the motions of proteins are a key component in understanding how the structure of a protein is related to its dynamic function." Since running these simulations is computationally expensive, they are currently performed by individual laboratories that have the resources to conduct this research. The purpose of this project is therefore to provide a data Grid of different simulations so that users will have a single point of access to this information.
Two members of the project participated in the study and were able to represent both a developer and a system user point of view. It should be noted that both this study and the DCOCE study were substantially shorter and only involved one workshop each.

3.4.2 Issues uncovered

People and Security

The first point that arose in this study was that the security roles of operatives in a system are frequently overlooked, and technical security mechanisms are generally assumed to solve a security problem. Many of the administrative duties in the system, such as backup, patching, maintenance of the authentication mechanism (in this case based on SSL digital certificates, and a username and password combination for users who do not have certificates), and maintenance of the authorisation mechanism (role-based access control) were not initially apparent.

Identifying these required detailed and probing questions, for example when the participants mentioned that the system was backed up (*"who backs the system up? Is there a policy for when and what to backup?"*), or that digital certificates were used to authenticate users (*"How do users get a certificate?", "Who do they apply to for access to the system?"*). Simply establishing that an administrator has to monitor, backup, and maintain the system – with little to no supervision or help – throws up a number of questions with regards to both the scalability of the system (can the tasks expected of the administrator be extended to cover one or two orders of magnitude more users?) and the effectiveness of the current system security (which in the absence of training, audit and documented policies is wholly dependent on the competence of the administrator – not on the technical countermeasures).

Stakeholder Conflicts

Issues of conflicting stakeholder requirements were also identified here. The question was asked: *"How important is it for you to be able to keep this [simulation data] secret?"* to which the answer was *"we have no need for confidentiality… At the moment."* When questioned further, *"from an academic user point of view, using your*

own words, how would you rate, how important would confidentiality be? Would it be low, unimportant, high, essential…" The answer was that the requirement for confidentiality was low, however from a *pharmaceutical company's* point of view, the requirement for confidentiality was deemed to be medium to high in some cases. However since the system did not currently involve pharmaceutical companies, the current requirement was originally judged to be low.

From a requirements point of view, capturing this information is important. On the one hand, the system *as it is* does not require that level of security, on the other, the system *as envisioned in a future development* may have a high requirement for this kind of security. Furthermore this also illustrates the need to identify and represent as many stakeholders in the system as possible to identify potentially conflicting viewpoints.
With regards to security, it is very important to understand the need for a cost-benefit analysis of any security decision. The differences between the short and long-term security needs in the system do not necessarily have to cause serious difficulties. It is cheaper to compromise on a short-term implementation than it is to compromise on the long-term design. Any security mechanisms that have been designed but not implemented will be cheaper to implement at a later date than in a system where it is necessary to overhaul the original design.

3.5 Digital Certificate Operation in a Complex Environment (DCOCE)

3.5.1 Overview

The DCOCE project is tasked with analysing the use of digital certificates and public key infrastructures (PKI) within the complex environment of Oxford University, with its (semi-) autonomous colleges and departments. The emphasis of the project is the use of X.509 digital certificates for authentication to services. It should be noted here that the purpose of the system is to provide a security infrastructure on which other services to users can be supplied.
A total of four stakeholders participated in the study, and each represented a different point of view. These consisted of:

- The university point of view. The view of the organisation that owns and manages the project.
- A developer of DCOCE. The view from the principal developer of DCOCE
- A user of the system. The view from an academic who will use DCOCE as a means of pursuing research
- A data provider. The view of an organisation that provides access to its data based on the authentication of staff and students through DCOCE.

3.5.2 Issues Uncovered

Stakeholder Views of Security

The study highlighted that issues of usability were of great concern, and the developer was keen to adopt a development approach that took stakeholder needs into account:

> *"The decision to look into PKI [(was taken]) and we're keen to go forward on that rightly or wrongly, and then we try and take it to our stakeholders and see how it fits and see what the needs are. So in a way we're kind of forced to go down the route of PKI, but we're trying to build something that will fit the stakeholders."*

Some of the challenges facing this project were identified from the university representative, such as:

> *"(The need to) have a system that is usable by my set of users at the university, that vary hugely in terms of their knowledge of IT and knowledge of security and knowledge of computing generally. (…) It needs to be scalable so you're issuing it to lots of new students that are arriving each year, so it's got to be scalable and dynamic and something that you can manage and keep on top of. Obviously it mustn't be too expensive, but that's part of the scalability issue."*

In addition to the user group, another important stakeholder group and their need for security was identified by asking what the performance measures of the system were:

> *"Where the performance really comes in, in terms of you need something that is secure enough to be trusted by the data service providers, that they're going to have faith in it, that it's working, that we're doing it properly. So we have to allow our users to access those resources, and for the data service providers to have confidence that our whole system is not so insecure that anyone in the UK or the world cannot get hold of credentials to log onto their system."*

Thus, security was essential for ensuring the trust of the data providers, and also the reputation of the university. A further complication with regards to managing the system was discovered when a question was asked about the logistics of the subscriptions to data service providers:

> *"Researcher: with the actual subscriptions with the data service providers, is it done on a university basis, or is it done on a more 'by department basis', so some departments would have access to these resources whereas others wouldn't, or if one department has it, the others wouldn't?*
> *University: That's a very good question actually. Within Oxford, if you want to have access, there's an agreement that you really should go through the systems and electronic resources department, so that these deals can be brokered like that. However one of the things that we were to find out was that there are some departments, there are some colleges that have got their own deals, so that was an interesting question for us. So yes they do do that."*

Usable Security

As has been stated, there was a need for ensuring that DCOCE provided usable security. The presence of a user in the case study provided a means of directly eliciting that stakeholder's point of view. When asked about current organisational policies for security, the user responded that the policies varied depending on location and who was managing the computers.

> *"Researcher: do you think that's a good thing, different policies, or would you like to see one policy that covers everything?*
> *User: I'd like to see one policy that covers everything. Because I have to do different things according to where I am*
> *Researcher: So is there a central place within the university which dictates policy for security, or is it up to the individual departments to set their own*
> *User: it's up to the different departments and colleges"*

Some usability information was uncovered by asking how much effort the user was willing to go to for the added security of the public key infrastructure of DCOCE:

> *"User:I don't want it to be any more difficult than it is already.*
> *Researcher: and how difficult is it*
> *User: Just username and password"*

It became clear that the priority of users is not security, but the ability to achieve their production tasks. This was further illustrated by the question:

> *"what sort of frequency of password change would you find acceptable?*
> *User: oh, I never change my email password…"*

Despite this lack of interest in actively pursuing security, the user was asked to assess the impact that a malicious attack on their personal data would have, highlighting the need for effective security:

> *"Researcher: so if somebody else got access to that data [user's research website] and changed it, would it have any impact on you or the work that you do?*
> *User: it would probably affect my credibility*
> *Researcher: Would that impact maybe on the university or the department or an outside agency*
> *User: well if my credibility slips that means that other people don't want to work or collaborate with me on research."*

The following section presents the analysis of the case studies, together with a model of the factors that affect the design process of a usable and secure socio-technical system.

4 Analysis

The model described in this section is the result of the grounded theory analysis (see section 3.1.2) of the transcripts of the case studies described in section 3.2 - 3.5. Transcripts were initially coded using the hermeneutics analysis package ATLAS.ti. With the help of this tool, these codes were then organised into categories. The categories consisted of **Motivation**, **Responsibility**, **Communication** and **Stakeholder**. Because

the findings from the analysis are too numerous to describe here, only the key points of interest from each category are presented in sections 4.2 - 4.5.

The final stage of grounded theory analysis consists of organising these concepts around a central category, and creating a 'storyline' explaining the resulting theory. This storyline is presented in section 4.6, and describes the fundamental factors that influence a participative socio-technical approach to security design. Throughout the following, high-level categories are highlighted in **bold** and detailed sub-categories are presented in *italics*.

4.1 Grounded Theory Model Semantics

All the models presented in the grounded theory analysis were generated using ATLAS.ti and consist of network diagrams that relate the different categories identified. The semantics of the notation are as follows:

- == means *is associated with*
- => means *is a cause of*
- [] means *is part of*
- <> means *contradicts*

A simple example of this is in the following diagram:

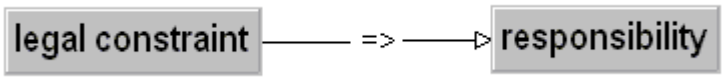

Figure 1: Example Grounded Theory Network Diagram

This diagram means that a "legal constraint" *is a cause of* "responsibility".

4.2 Responsibility

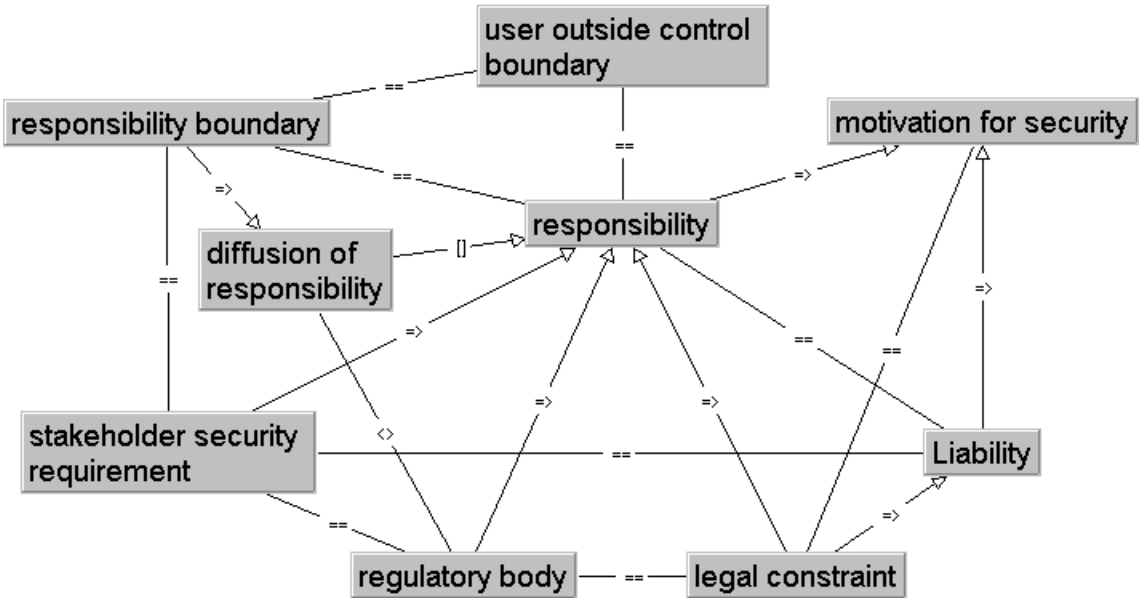

Figure 2: Responsibility Model

As shown in Figure 2, **responsibility** for security is tightly linked with **motivation**, *liability*, and *stakeholder security requirements*.

4.2.1 Boundaries of Responsibility and Control

Identifying the roles of people in the system and the environment in which they operated generally led to the identification of boundaries:

1. of *control*
2. of *responsibility*

> *"Researcher:(…) do you assume that there is an admin at each provider?*
> *Stakeholder3: yes it is assumed that there will be somebody who has the role of an admin, whether that is there…*
> *Stakeholder2: I believe they're outside EGSO*
> *Stakeholder3: yes they're outside EGSO*
> *Stakeholder2: they don't have to know the nitty gritty*
> *Researcher: but you're assuming that someone at the provider end is in charge of the resource and is capable of modifying access and things like that."*

It is important to note that **responsibility** does not necessarily go hand-in-hand with *control*. In this example, since the provider administrators operated outside EGSO, they weren't inside the *control boundary* of EGSO – i.e. the boundary within which a component (human or technical) can be made to behave in a specified manner. This led to the implication that they were also outside the security *responsibility boundary* (i.e. the boundary within which security should be addressed by the project) and therefore should not be included in the security design process. As seen here, the process of identifying these boundaries can also uncover some evidence of *diffusion of responsibility*.

In situations where there is a lack of *control* over parts of the system, yet security has to be provided, the **stakeholders** expressed the need for making security as lightweight as possible. This can be seen as an indication that low security overheads and easy-to-use security mechanisms are important for situations where *control* (i.e. enforcement, monitoring, or auditing for example) is not possible. This is a key argument in favour of usable security since it provides a means of addressing the lack of control of security.

4.2.2 Diffusion of Responsibility

Another property of **responsibility** that has been identified is the propensity to assume that another party will or should take care of security. This is what social psychologists call *diffusion of responsibility*: the notion that everyone assumes that someone else will take care of a particular problem [Darley & Latané, 1970]. This has been identified at the system level through undocumented assumptions (e. g. that administrators will take care of maintaining access control lists, backup systems and perform special services).
It has also, in some cases, been identified at the project level:

- Where lack of control over a component (as identified in section 4.2.1) leads to a reaction of avoiding the consideration of security for that component.
- Where some security issues are assumed to be the province of another party (namely governing bodies such as Gridstart, or the e-Science Security Task Force). For example, in order to use certificates in the system, a network of trust between certification authorities has to be in place.

  "It really shouldn't be up to EGSO to establish this network of trust itself. It should be relying on people to certify people within countries and organisations."

As seen in this quote, the **responsibility** for setting this up and administering it was argued to be in the hands of a third party, possibly the governing bodies of the e-Science projects – but no discussion had arisen between the project and the governing bodies with regards to resolving this issue.
It is important to note that *diffusion of responsibility* occurred only in areas where **responsibility** was not clearly assigned. In the case of CLEF, the legal and ethical frameworks ensured that the project took **responsibility** for security; therefore it is not surprising that there was no evidence of *diffusion of responsibility* at the project level in this study.

4.3 Motivation

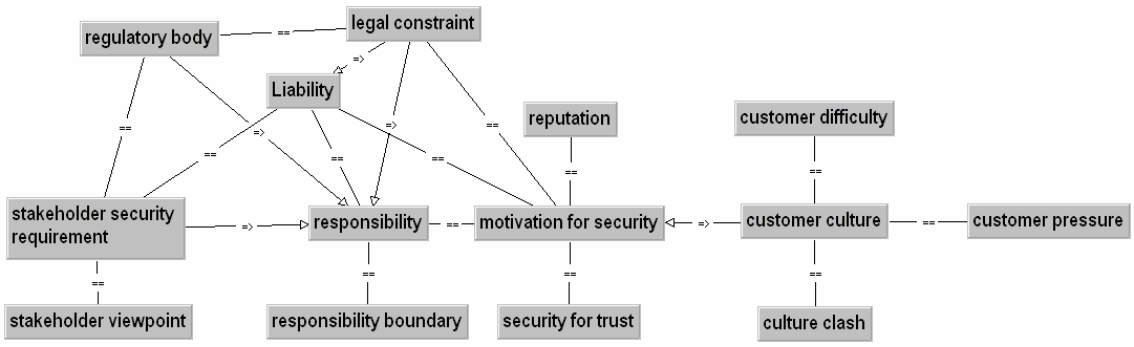

Figure 3: Motivation Model

During the studies, the **motivation** to apply security varied from enthusiastic to 'not interested'. It was stated, for example, that security would be addressed once functionality was finished. In one study, this led to parts of the development team in the project that was not involved in the workshops ignoring any matters pertaining to security (even when part of the project *was* addressing security issues, the rest of the project tended to be completely uninterested, e.g. "*He's* [the architect] *not 'all right, what's been happening with the security, what's coming out of it?' do you know what I mean?*").

A detailed analysis shown in Figure 3 describes all the factors that affect the **motivation** for security during the development process. These are:

- **Responsibility**
- *Liability*
- *Reputation*
- *Trust*
- *Customer culture*.

Responsibility is a key motivator for addressing security. In the EGSO case study, the manager was keen to take responsibility for security (even though he was not explicitly given this responsibility) and facilitated and participated in the security design process. In the CLEF and DCOCE studies, **responsibility** for security was squarely with the participants and therefore strongly motivated them to address the issues.

Liability is a refinement on the notion of **responsibility** in that it represents what a third party expects to be the responsibility of the project – not necessarily what the project recognises or accepts as their responsibility. Liability is a motivator for security in the sense that if the project were to facilitate or inadequately guard against an attack that damaged a **stakeholders'** assets, the project would have to face a cost. This could be financial cost, but more importantly the *reputation* of the project would be affected.

Safeguarding the *reputation* of the project is a particularly potent motivator for security, since the system depends on the goodwill of providers to operate. Should the project's *reputation* become tarnished, it is possible that customers and users would no longer *trust* the system to run on their machines, thereby putting the whole survival of the project at risk. The need for safeguarding this trust is a strong motivating factor for addressing security.

Finally, *customer culture* can also affect the **motivation** for security:

> "*They (customers) want something that they can sit down and physically play with, rather than something which is presented on paper. (…) They'd be happy with code, whether it works or not, they'd be happier with seeing some code rather than seeing some abstract representation of some high-level app…*"

As illustrated by the quote, the customer culture in this case is perceived by the participants as placing a great deal of emphasis on achieving functioning prototypes as quickly as possible, without necessarily going through structured engineering approaches. The pressure is therefore put on the developers to provide functionality as quickly as possible, to the detriment of security.

4.4 Communication

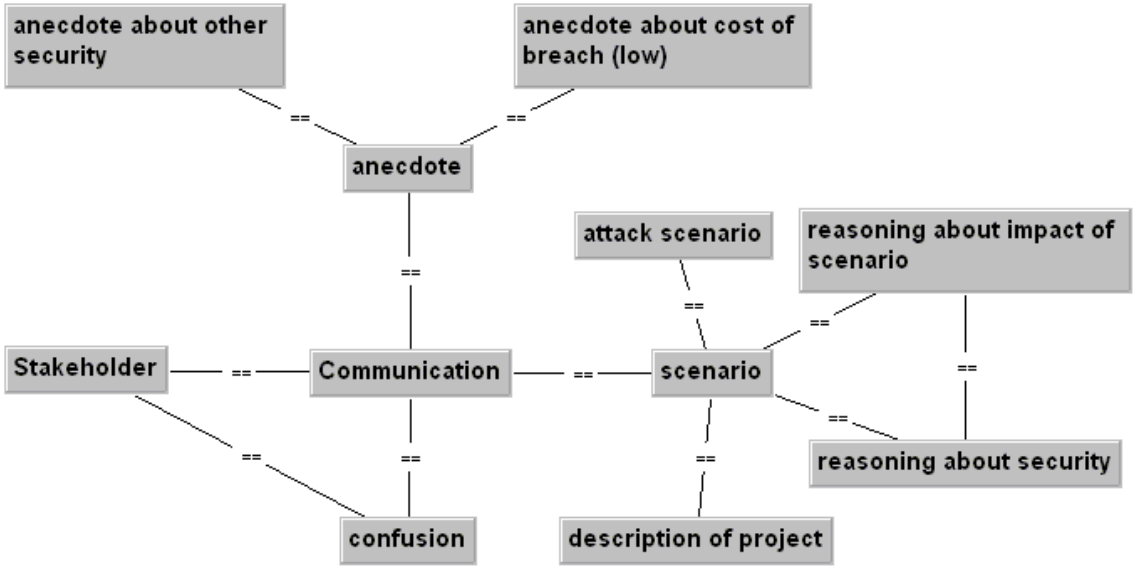

Figure 4: Communication Model

Confusion

Initially, participants required detailed explanation of the security concepts because of their extremely precise and abstract nature. During the discussion the differences between integrity, confidentiality and availability could become confusing, particularly after discussions had been ongoing for long periods of time.

> *"Researcher: what about availability? Availability of this program. Judging from what you've said I don't think that availability is that…*
>
> *Stakeholder2: should it not match the most important resources?*
>
> *Researcher: that's an interesting question… I think it's a bit beguiling, because we're not seeing the picture as it is…. No no no I'm sorry my mistake, I think availability is high, is a very high level, you're right… I'm just a bit confused."*

This is made somewhat worse by the fact that dependencies between assets can link two different security concepts in two different assets. For example the integrity of server software can directly affect the availability of the data served by that machine, or the confidentiality of a user's activities.

Addressing issues of confusion, particularly for the purposes of eliciting the security requirements of each asset of the system, became very important. During the studies, the participants were finding it difficult to assess the importance of the security property of an asset (e.g. the integrity of server software), therefore the question was clarified by using a *scenario* in which this particular property was compromised and asking the participants to rate how damaging this would be to the system.

The decision to use *scenarios* in the security requirements elicitation came from the observation that the **communication** of more complex security concepts throughout the

case study generally took place in the form of *anecdotes* and *scenarios* which will be described in the following sections.

The Need for Scenarios

To overcome the complexity associated with abstract security concepts, *scenarios* were discussed extensively in the case studies. As well as being used to elicit security requirements, *scenarios* were used to propose potential threats, suggest design solutions and describe how these would behave.

In more complex discussions, *scenarios* were also supported through the use of simple graphical representations (on a white board for example). These allowed the clarification of complex **communications** between different participants, and ensured that people did not talk at cross purposes.

It should be noted that abuse cases [McDermott & Fox, 1999] are a form of documented *scenario* for the purposes of identifying security requirements. The difference here is that abuse cases are directly used to identify vulnerabilities by modelling attacks, whereas in this case *scenarios* are used to elicit information from the participants. As such these *scenarios* are not a security analysis tool like abuse cases, but a **communication** tool which then supports security analysis through the participation of others. Despite this difference, the notation employed by abuse cases (namely UML use cases) could easily be used to document *scenarios*.

The Role of Anecdotes

Knowledge about security is generally expressed in the form of *anecdotes* – either personal or vicarious stories. In the case studies, security *anecdotes* were used by participants to explain and justify the possibility or likelihood of a particular security problem. For example:

> *"Stakeholder1: well yes again, X was telling me the other day that he's got some files that have vanished, and it's because he was running a mirror.*
>
> *Stakeholder2: mirror?*
>
> *Stakeholder1: he had a mirror running that should have been stopped, somebody deleted some data at the other end, so his data disappeared as well because he had got the delete switch on. Now this is the problem with the, not necessarily with the administrator, directly doing something, but inadvertently they allowed a change in the system at the other end to affect the copy at your end."*

Anecdotes about past attacks gathered from **stakeholders** who have experience in the field can also be a useful source of information. In the absence of other risk measurements, *anecdotes* can serve during the risk analysis as a means of informing the estimation of the likelihood of an attack occurring. Whilst this is not an ideal solution, it is frequently the case that there is no other information available on which to base this estimate.

4.5 Stakeholders

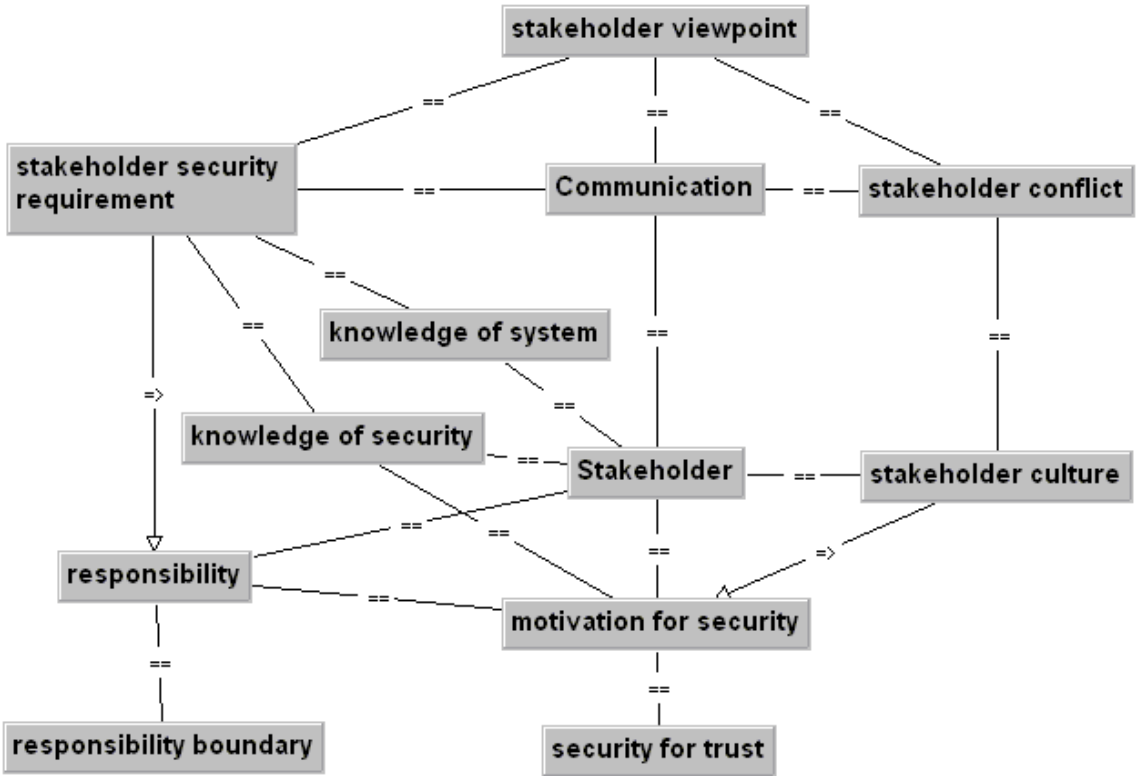

Figure 5: Stakeholder Model

4.5.1 Stakeholder Viewpoint

The biggest benefit of involving **stakeholders** in the security process was that it was possible to directly elicit their point of view. This provided very rich information about security needs, constraints and limitations that would be acceptable to the different **stakeholders**. As an example, other than lack of control, another factor driving the need for data providers to have easy-to-use security was the stated need for low buy-in. This was necessary in order for the project to secure as many providers as possible, thereby creating value in their system.

It is interesting to note that even with a relatively small set of **stakeholders** participating in the process it is possible to identify points of view from a variety of other **stakeholders** as related through the participants. These points of view can serve as a basis for identifying the value of the security properties of the system's assets. In EGSO for instance, it was initially stated that there was no need for confidentiality of the solar data. When asked about the point of view of the organisations supplying the data, it was identified that some solar data providers did have a requirement for temporarily ensuring the exclusive access to their data.

4.5.2 Stakeholder Knowledge

Different **stakeholders** have different types of knowledge that are relevant to the system design:

- System knowledge
- Security knowledge

Stakeholders have a different understanding about diverse areas of the system. Developers for instance are particularly focussed on the technical needs and possibilities of the system. System users are more interested and knowledgeable in the areas of application of the system and how the functionality provided will be useful. Using this knowledge in system development is traditionally the province of requirements elicitation, where the needs of the **stakeholders** are captured in order to inform the design of the system.

Whilst it is typical in system development to gather functional requirements from **stakeholders**, security development tends to adopt a different approach. General approaches to security derive security requirements from checklists, risk assessments based on questionnaires or an analysis of the system in order to determine what security is necessary. This type of approach only makes use of security experts' security knowledge, and whilst they have the most security knowledge of all the **stakeholder** groups of the system, they are not the only source of security knowledge or needs. The needs of users, data providers, administrators, and developers are also important for security. As such the knowledge of these other **stakeholders** is also particularly relevant in the identification of security requirements that reflect accurately what the **stakeholders** want.

It is also important to realise that *stakeholder knowledge* of security can be limited or even flawed. Mistakes, preconceptions and misunderstandings can affect the direction of the discussion. As an example, a load balancing mechanism for EGSO was proposed as a means of addressing denial of service attacks. Although this mechanism would be useful under heavy but normal operation, it would not be particularly effective against a targeted attack. *Stakeholder conflict* (see section 4.5.4) can arise out of mistakes, however in the case of mistakes or misconceptions, this can usually be resolved through **communicating** the reasoning behind different positions.

4.5.3 Stakeholder Security Awareness

The only previous attempt to involve **stakeholders** directly in the security process has been reported by [James, 1996]. Involvement there was limited to the security planning of information security in an existing organisation – not in the development of a technical system.

James found that awareness of security needs was raised after the involvement of the participants. In these case studies there is also evidence that involving **stakeholders** in this process has raised awareness. In the words of one participant who was paraphrasing a politician at the time:

> *"… what [AEGIS] is doing is reducing the unknown unknowns and converting the unknown unknowns into known unknowns."*

Essentially, the participative nature of the AEGIS process allowed the participants to become more aware of security issues which had previously not been known about.

4.5.4 Stakeholder Conflicts

This category serves to illustrate the cases when different **stakeholders** do not agree. In order to move on from conflicts, it becomes necessary to reach an understanding with the two (or more) **stakeholders** regarding the disagreement. Either the disagreement arises out of incomplete knowledge (such as ignorance of a particular threat, for example) or it arises out of genuinely differing but equally valid points of view.
If these conflicts occur, they can usually be resolved after both sides have argued their position, and possible solutions can then be explored. And whilst situations where this fails to achieve a solution were not encountered in our studies, it is likely that they could. This would probably mean that both sides had equally valid viewpoints, and such a conflict would be very useful in uncovering a serious issue that might require a higher-level strategic decision.

4.6 Model of factors and issues in socio-technical security design in e-Science

A model of these factors is presented in Figure 6. It should be noted that whilst these factors were identified from information gathered from the application of AEGIS, efforts were made to ignore the issues that were the direct result of AEGIS (for example modelling assets, or eliciting security requirements based on these assets). Instead the focus was put on identifying the general factors that *affected* the socio-technical security design process, thereby providing a better understanding of the more general act of designing security.

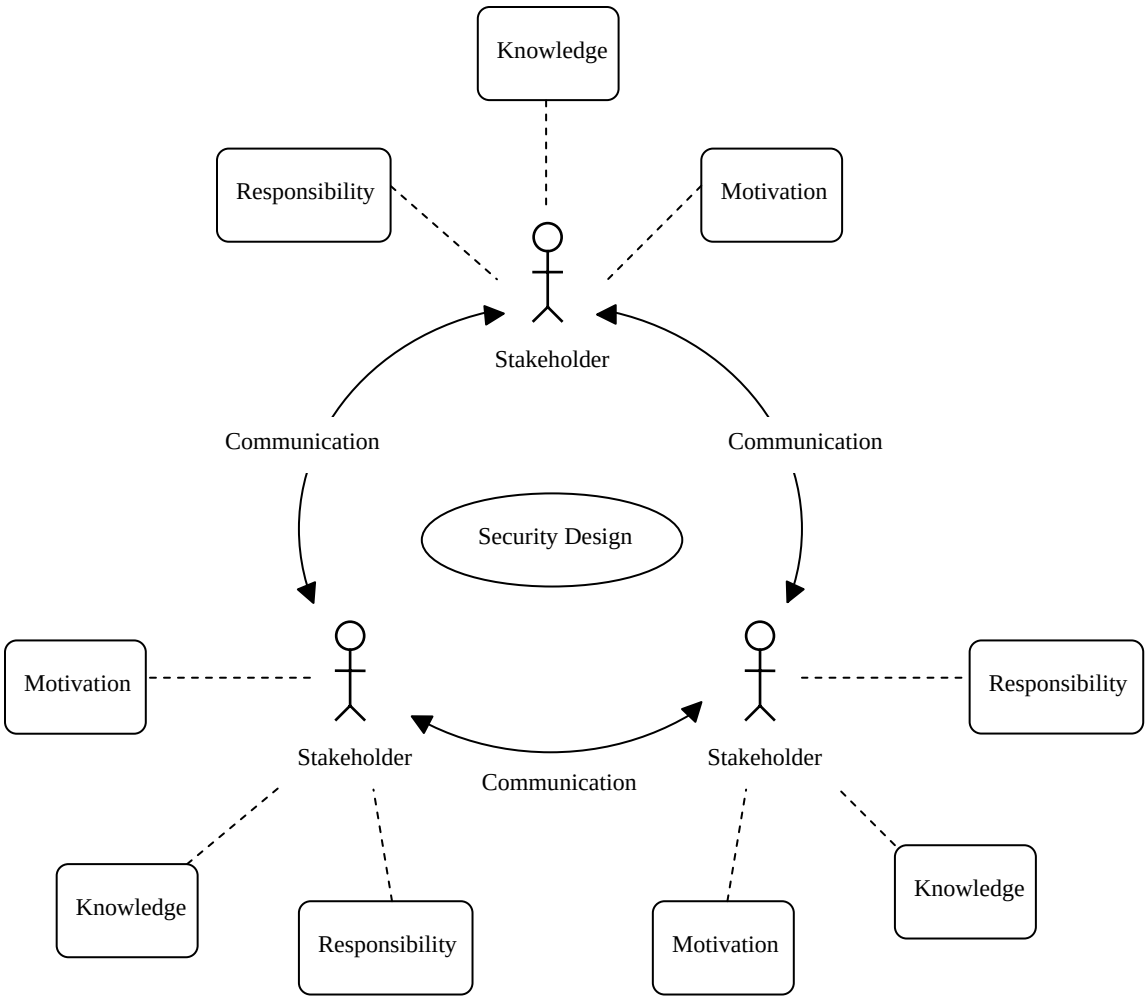

Figure 6: Grounded Theory Model of Socio-Technical Secure System Design

The aim of security design is to create a system that is adequately protected from undesirable events in that system. It is important for the success of the security that relevant knowledge be present during the design process. This includes the need for accurate knowledge about:

1. the system
2. security practice

Knowledge about the system refers to information that describes what the system is intended to do, how it should operate and any other information about the system relevant to the design of security. The best source for this information exists in the knowledge of **stakeholders** such as users, developers, system owners, system administrators, etc.

Knowledge about security practice refers to security concepts and principles, the understanding of the need for security to be usable, insight into threats, vulnerabilities and risk, etc. This type of information exists in the knowledge of **stakeholders** such as security experts, but also as part of the security design process itself, which informs and directs the act of designing security. In the case studies, the AEGIS process provided some knowledge about security practice, whilst also recommending the involvement of

security experts as a means of providing the essential knowledge gained through experience.

Given that any **stakeholder** (for example a user or a security expert) has a degree of knowledge about both the system and security practice, the model does not distinguish between the extent and type of knowledge that different **stakeholders** have.

In order to ensure that this knowledge is available, it is necessary for all the relevant **stakeholders** in the system to be identified and represented in the design process. These **stakeholders** include people who are more knowledgeable about the system – for example users, developers or administrators – and people who are more knowledgeable about security practice, namely security experts. This is particularly important for e-Science which aims to bring many different **stakeholders** together in distributed environments and where the **communication** between **stakeholders** can be more limited.
One of the best ways of achieving this representation is to ensure the participation of **stakeholders** in the process. However, each **stakeholder** has different security responsibilities (which range from significant in the case of security experts to very low in the case of some users) and various levels of **motivation** to address security needs. As was seen in one case study, this can result in certain **stakeholders** declining to participate in the security design, either through a perceived lack of **responsibility** (for example through *diffusion of responsibility*) or insufficient **motivation** (for example with the need to achieve functionality taking precedence). Despite this, ensuring that the relevant viewpoints of **stakeholders** in the system are represented in the security design is critical in making sure that the final design addresses the needs and requirements of these **stakeholders**.
The issues of **motivation** and **responsibility** can also affect the security design process beyond determining whether **stakeholders** decide to participate or not. That is to say that issues of *responsibility boundaries*, in other words the perception of the limits of **responsibility** of a particular **stakeholder**, also limit the extent to which **stakeholders** decide to consider security needs. In practice, this can be seen when **stakeholders** decide not to address specific issues because they perceive the problem to be the **responsibility** of another party. The problem arises here when there is no **communication** between the two parties and both of them assume that the issue is the **responsibility** of the other (another instance of *diffusion of responsibility*). The distributed nature of many e-Science development environments hinders clear **communication**s and makes such examples of *diffusion of responsibility* far more likely.

The process of socio-technical security design is fundamentally a **communication** exercise between the different **stakeholders** in the system. Without effective **communication** the relevant design information – such as requirements, constraints or necessary discussions – would be impossible. As a result, the primary purpose of a socio-technical security design process is as a means of *facilitating* this **communication**.

The concept of **communication** was very strongly tied in practice to *anecdotes* and *scenarios* as means of expressing security knowledge and reasoning about security. *Anecdotes* were frequently used to communicate knowledge about real security issues.

Scenarios, however, were even more widely used as a means of communicating security concepts, reasoning about security principles and justifying points of view. Given that participation in security design has not been extensively researched, it is useful to note the effectiveness of *scenarios* at communicating specific and detailed technical information to non-technical **stakeholders** (such as users).

Successful **communication** requires that **stakeholders** share their knowledge and points of view with as little bias and as few misunderstandings or *confusion* as possible. This is necessary to ensure that the design process remains focussed on matters of security design, as opposed to having to address issues of semantics or other unrelated concerns. The aim is to ensure that *stakeholder conflicts*, in other words disagreements, are related to genuine and valid opposing points of view, as opposed to differences that arise out of miscommunications or *confusion*. Resolving these genuine disagreements is a key aspect of the act of designing security.

5 Conclusions

For e-Science, the need to devise security mechanisms that address stakeholder concerns, and are both effective and usable is a significant concern. As seen from the case studies, involving stakeholders in the design of security provides a very effective means of identifying their needs. In addition, the presence of stakeholders during security design also provides additional benefits in raising awareness and knowledge of security issues in the system. Finally, understanding stakeholders' capabilities also facilitates the design of appropriate countermeasures, making the final system well-suited to its intended users.

The grounded theory model of the factors and issues surrounding socio-technical secure system development provides a useful theoretical framework that can be used to analyse the reasons for the success or failure of a given socio-technical design method. This can be useful for future research as a means of evaluating other secure system design methodologies and identifying some of their strengths and weaknesses, together with proposing areas in which security design methodologies can be improved.

In addition to helping to evaluate other methodologies, the factors identified in the model have implications for secure system design in general. **Motivation** and **responsibility** are two key aspects that extend beyond the scope of a design methodology. Specifically assigning **responsibility** to individual **stakeholders**, or ensuring their **motivation** to address the security issues, is not in the power of a design methodology to enforce. However, it can be a strong recommendation that one of the first steps any security design exercise should take is to ensure the clear assignment of **responsibility**. This is necessary to ensure that everyone in the project knows who is in charge of security, and addresses the problem of *diffusion of responsibility*. A final point is that responsible **stakeholders** that do not have the authority to implement security decisions will encounter significant difficulties. As a consequence, **responsibility** must either be assigned to people who have authority (senior management for example), or authority must be given to those who are charged with security.

The assignment of **responsibility** also has an impact on the **motivation** of **stakeholders** – namely increasing it in those who are **responsible** for ensuring security. Maintaining the momentum and **motivation** for security can be a difficult task when faced with competing demands (e.g. functionality, cost, time-to-market), and the complexity of security design. An initial **motivation** for achieving security drives the desire to adopt a secure system design methodology. It is therefore important for security practice to understand the motives behind the need for security and address these. The motives can relate to legal requirements, fears of exposure to attack, or having valuable assets. Clearly addressing these underlying issues should be a key element of any security design methodologies if they are to keep in touch with the original **motivation** for security. Furthermore, the process of security design should be engaging, inclusive and understandable to all the participants in order to avoid discouragement. Motivational and organisation psychology are ideal fields of research from which further insight into these issues could be gained.

The identification of *anecdotes* and *scenarios* in secure system design has more practical and immediate implications for security design in general. One of the key elements of any design exercise is ensuring the good **communication** of the participants. Security methods that adopt and support *scenarios* are much more in tune with the way people tend to communicate about security. As such they have a greater chance of being understandable, facilitating the clear **communication** between **stakeholders**. This is not to say that *scenarios* should be the only means of modelling or reasoning about security. Some of the weaknesses of *scenarios* are related to the difficulty in generalising their information content. That is to say that *scenarios* are highly specific descriptions of particular events in a system, whereas security needs have to encompass the system as a whole. *Scenarios* should therefore be used in conjunction with other security analysis techniques, as a particularly useful method for explaining security concepts and reasoning.
Similarly, *anecdotes* used in security design discussions are not necessarily accurate or representative of the problem space. However, *anecdotes* have the benefit of being a persuasive source of security knowledge during these discussions. In the eyes of many, information related from *anecdotes* holds the advantage of having actually occurred, as opposed to being a theoretical possibility. Simply relying on *anecdotes* as the sole source of knowledge informing security design is risky and prone to error. However used in conjunction with other sources of knowledge, *anecdotes* have the benefit of being a useful source of information that can be easily communicated, understood and accepted by different **stakeholders**.

6 References

Adams, A. & Sasse, M. A. *Privacy in Multimedia Communications: Protecting Users, Not Just Data.* In A.Blandford, J.Vanderdonkt & P.Gray [Eds.]: People and Computers XV - Interaction without frontiers.Joint Proceedings of HCI2001 and ICM2001, Lille, Sept.2001 2001. pp 49-64. Springer.

Adams, A. & Sasse, M. A. *Users Are Not The Enemy.* Communications of the ACM 1999. Vol. 42, No. 12 December

Backhouse, J. & Dhillon, G. *Structures of responsibilities and security of information systems.* European Journal of Information Systems 1996.

Barki, H. & Hartwick, J. *Rethinking the Concept of User Involvement.* MIS Quarterly, March 1989. pp 53-63.

Baskerville, R. *Investigating Informations Systems with Action Research.* Communications of AIS 1999. Volume 2, Article 19

Beyer, H. & Holtzblatt, K. *Contextual Design : Defining Customer-Centered Systems*. 1998. Morgan Kaufmann Publishers, Inc.

Butler, S. *Security Design: Why It's Hard To Do Empirical Research.* 2000. Workshop on Using Multidisciplinary Approaches in Empirical Software Engineering Research, affilated with the 22nd International Conference on Software Engineering (ICSE 2000). http://www-2.cs.cmu.edu/~Vit/paper_abstracts/secure.butler.html

Butler, T. & Fitzgerald, B. *A case study of user participation in the information systems development process.* Proceedings of the eighteenth International Conference on Information Systems 1997.

CLEF. *Clinical e-Science Framework.* 2004. http://www.clinical-escience.org/

Darley, J. M. & Latané, B. *Norms and normative behaviour: field studies of social interdependence*. Altruism and Helping Behaviour. 1970. New York: Academic Press. J.Macauley & L.Berkowitz (eds).

Department of Trade and Industry. *Information Security Breaches Survey.* 2006. http://www.pwc.com/uk/eng/ins-sol/publ/pwc_dti-fullsurveyresults06.pdf

Dhillon, G. & Backhouse, J. *Current directions in IS security research: towards socio-organizational perspectives.* Information Systems Journal 11 2001. pp 127-153.

EGSO. *European Grid of Solar Observations.* 2005. **http://www.egso.org**

Flechais, I., Mascolo, C., & Sasse, M. A. *Integrating Security and Usability into the Requirements and Design Process.* Proceedings of the 2nd International Conference on Global e-Security 2006.

Flechais, I., Sasse, M. A., & Hailes, S. M. *Bringing Security Home: A process for developing secure and usable systems.* New Security Paradigms Workshop 2003.

Gorlenko, L. *Small world, water coolers, and the challenge of remote collaboration.* Interactions 2005. 12 (6) , pp 42-44. New York, NY, USA, ACM Press.

Hitchings, J. *A Practical solution to the complex human issues of information security design.* Proceedings of the 12th IFIP TC11 international conference on information security 1996.

Irestig, M., Eriksson, H., & Timpka, T. *The Impact of Participation in Information System Design: A Comparison of Contextual Placements.* Proceedings Participatory Design Conference, Toronto Canada 2004.

James, H. L. *Managing Information Systems Security: a Soft Approach.* Proceedings of the Information Systems Conference of New Zealand 1996. IEEE Society Press, Los Alamitos, CA.

Ka-Ping, Y. *User Interaction Design for Secure Systems.* 2002. http://zesty.ca/sid

Martin, P. Y. & Turner, B. A. *Grounded Theory and Organizational Research.* The Journal of Applied Behavioural Science 1986. (22:2) , pp 141-157.

McDermott, J. & Fox, C. *Using Abuse Cases for Security Requirements Analysis.* Proceedings of the 15th Annual Computer Security Applications Conference 1999.

Mumford, E. *Designing Human Systems - The Ethics Method*. 1983. **http://www.enid.u-net.com/C1book1.htm**

Mumford, E. & Weir, M. *Computer Systems in Work Design - The ETHICS Method.* 1979. Associated Business Press.

Rousseau, D. M. *Managing the Change to an Automated Office: Lessons from Five Case Studies.* Office: Technology & People 1989. 4 , pp 31-52.

Saltzer, J. H. & Schroeder, M. D. *The protection of information in computer systems.* IEEE 1975.

Sasse, M. A. & Flechais, I. *Usable Security: What is it? How do we get it?* Lorrie Faith Cranor & Simson Garfinkel [Eds.]: Security and Usability: Designing secure systems that people can use 2005. pp 13-30. O'Reilly Books.

Siponen, M. & Baskerville, R. *A New Paradigm For Adding Security into IS Development Methods.* Eighth Annual Working Conference on Information Security Management & Small Systems Security, Las Vegas, Nevada, USA 2001.

Siponen, M. T. *An Analysis of the Recent IS Security Development Approaches: Descriptive and Prescriptive Implications.* Information Security Management: Global Challenges in the New Millennium 2001. pp 101-123. Idea Group Publishing.

Straub, D. W. & Welke, R. J. *Coping with Systems Risk:Security Planning Models for Managerial Decision-Making.* MIS Quarterly 1998. 22:4 , pp 441-469.

Tait, P. & Vessey, I. *The Effect of User Involvement on System Success: A Contingency Approach.* MIS Quarterly, March 1988. pp 91-107.

Whitten, A. & Tygar, J. D. *Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0* . Proceedings of the 8th USENIX Security Symposium, August 1999, Washington 1999.

Wong, E. Y. W. *A Study of User Participation in Information Systems Development.* 1994. **www.is.cityu.edu.hk/Research/WorkingPapers/paper/9405.pdf**

Zurko, M. E. & Simon, R. T. *User-Centered Security.* New Security Paradigms Workshop 1997.