2017

Contextualizing Secure Information System Design: A Socio-Technical Approach

Abdul Rahim Charif
Nova Southeastern University, ac1835@nova.edu

NSUWorks Citation

Abdul Rahim Charif. 2017. *Contextualizing Secure Information System Design: A Socio-Technical Approach.* Doctoral dissertation. Nova Southeastern University. Retrieved from NSUWorks, College of Engineering and Computing. (999) https://nsuworks.nova.edu/gscis_etd/999.

Contextualizing Secure Information System Design: A Socio-Technical Approach

By

Abdul Rahim Charif

A Dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
Information Assurance

College of Engineering and Computing
Nova Southeastern University

2017

An Abstract of a Dissertation Submitted to Nova Southeastern University in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Information Systems

Contextualizing Secure Information System Design: A Socio-Technical Approach

by

Abdul Rahim Charif

Secure Information Systems (SIS) design paradigms have evolved in generations to adapt to IS security needs. However, modern IS are still vulnerable and are far from secure. The development of an underlying IS cannot be reduced to "technological fixes" neither is the design of SIS. Technical security cannot ensure IS security. Generations of SIS design paradigms have evolved, all with their own sets of shortcomings. A SIS design paradigm must meet well-defined requirements, yet contemporary paradigms do not meet all these requirements. Current SIS design paradigms are not easily applicable to IS. They lack a comprehensive modeling support and ignore the socio-technical organizational role of IS security. This research introduced the use of action research in design science research. Design science paradigm was leveraged to introduce a meta-design artifact explaining how IS requirements including security requirements can be incorporated in the design of SIS. The introduced artifact CSIS provided design comprehensiveness to emergent and changing requirements to IS from a socio-technical perspective. The CSIS artifact meets secure system meta-design requirements. This study presented a secure IS design principle that ensures IS security.

Acknowledgments

I wish to thank my supervisor for his important comments concerning the dissertation and for his support to my research endeavors. I would like to thank my parents, for their immense support throughout this PhD research process. Gratitude also goes to my colleagues and friends, whose support I have always appreciated.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1. Introduction

The advances in network communication introduced growing use of global communications in information systems (IS). The increased dependency on computer communications and IS increased the threat of incidents related to information security. According to the U.S. Information Security Oversight Office, estimated costs associated with information security were $5.2 billion in 2010 (ISOO, 2015). ISOO reports indicate that information security continues to be the main driver of all IS costs. Cybercrime continues to cost billions of dollar to organizations. The importance of this research stems from this ongoing and well-known issue in IS security research; organizations continue to lose substantial amounts of money due to failure in securing IS. According to the most recent US cybercrime survey (Burg et al., 2014), out of the 500 respondents who did report their losses, the average annual monetary loss was approximately $415,000 for each organization. A great increase compared to prior years (Peters, 2009). In 2014, three in four respondents to the US State of cybercrime survey detected a security event in the past 12 months, and more than a third said the number of security incidents detected increased over the previous year.

In response, to ensure the security of information systems, researchers and practitioners have created different secure information system (SIS) design methods. Siponen et al., (2006) called for additional research as the review of existing methods indicate the need

for development of theoretically and empirically grounded SIS design methods. The existing SIS design methods are theoretically underdeveloped and lacking "*serious research*" (Baskerville, 1992) or filled with "*nonsense*" (Olnes, 1994 p. 628).
Siponen et al., (2006) contended that as the design theory approach has proved successful in prescribing better design processes for emergent classes of systems, reframing development of SIS design methods as a design science activity is a promising way to advance past this problem.

Siponen et al., (2006) defined six meta-design requirements to be met for a secure information system design. Siponen et al., (2006) asserted that the design of a secure information system cannot be reduced to technological fixes. Different SIS approaches cover different levels of IS but the models lack the comprehensiveness needed (Siponen, 2002). The overall aim of this research was to produce a SIS design method that is theoretically and empirically grounded. In pursuing this aim, this research developed a design artifact that includes security as a functional requirement in all stages of system design. The research problem and the research argument is described in Section 1.1. Importance of investigating the research problem is argued in Section 1.2. The definitions adopted in this study for the purposes of clarity and consistency of terms are described in Section 1.3. Literature review of secure IS design is discussed in chapter 2. The design science research process model is discussed in chapter 3 (Peffers et al., 2008). Action research and research findings are discussed in chapter 4. Finally, concluding the study in chapter 5.

1.2. Research Problem and Argument

Too often, information security is an afterthought in organizations. Organizations spend time and effort on designing and implementing security solutions for their IS as a repercussion of implementing the IS itself. However, these efforts are not as effective since security design aspects were not considered during the initial IS design and due to a lack of proper understanding of organizational context. The existing SIS design methods, at least the ones that can be integrated easily into IS or software development process have a narrow technical focus ignoring organizational aspects (Siponen, 2002; Siponen et al., 2006).

Other methods that take a socio-technical stance towards the organizational role of IS security e.g. viable systems (Karyda et al., 2001) suffer from applicability issues. Such methods cannot be integrated into IS development process. The need for a SIS design method that provides comprehensive modeling of technical and formal aspects of IS is being felt and is becoming more evident in literature (Siponen et al., 2006). Despite the wealth of literature in IS security, emergent SIS design approaches continue to fail in securing IS in organizations.

Information security design aspects have been ignored in IS development methods. In the literature, there has been a call for a SIS design method that goes beyond implementation of technical controls and includes organizational aspects (Siponen, 2002; Siponen et al., 2006). The research problem investigated in this study is how to design a secure information system. Current design methods promote the idea that IS security must be included in all IS development stages including design stages (D'Aubeterre et al., 2008). However, such modeling does not consider the context at which the organization operates under and thus does not provide design comprehensiveness (Siponen et al., 2006). This research argued that in order to secure the processes of information system in an organization, it is fundamental to understand the context of organizational processes and activities. Capturing the context of processes and activities within IS is essential to understand IS security requirements.

This dissertation contended that in order for secure information system design to be successful, a proper understanding of the context of the underlying information system activities and business processes is essential. Information systems security research needs to mature from mechanistic tendencies and ground itself in contextualism (Tejay, 2008). When considering the relevance and applicability of a given approach, one should bear in mind the context of the organization and environment where an approach will be applied. Using the context of activities to evaluate IS security allows for more control over IS. Therefore, incorporating the context of activities and processes when defining security requirements ensures the security of the information system design and helps incorporate security in all IS design stages.

The challenge for organizations lies in the ability to include IS security design aspects in the IS development process. This research study extended the existing body of knowledge in SIS design in three areas. First, this research study addressed a gap in the existing research by developing a theoretically grounded and comprehensive approach on how to design a secure IS that can be integrated into all IS development stages. Second, this research study investigated how to integrate socio-technical aspects in IS development stages using the context of processes in IS and organization. Finally, there has been limited research in examining the dynamics of access control during security requirement analysis and design stages (D'Aubeterre et al., 2008). This research included IS security requirements along with access control dynamics in all IS design stages.

1.3. Importance of Research Problem

All SIS design methods come with their own shortcomings. Siponen (2002) outlined the evolution of SIS design methods in literature in terms of their influencing disciplines: "IS security, computer security, practitioners, database security, and cryptology". Siponen (2002) also extended the generational classification of IS design approaches originally proposed by Baskerville (1992) to include disciplines of security (see Appendix A).

Early SIS design generations "naturalistic-mechanistic" were mainly advocated by practitioners and computer science researchers. Such approaches are means oriented, they are aimed at finding out what can be done, or what should the organization do with the help of available technical solutions. These design approaches fail to ensure the security of IS because they only focus on technical aspects of SIS design and do not take IS security requirements as a point of departure during IS design. Early SIS design generations failed to provide comprehensive security coverage to all aspects of system design.

Later generations were focused on IS modeling aspects. Unlike mechanistic methods, this generation considers organizational requirements during IS design. However, they focus heavily on business process modeling and information modeling. The objective in this approach is to extend security issues into workflow and business process management. Rohm et al., (1998) and Herrmann & Pernul (1999) suggested a layered framework according to which the security (mainly confidentiality and integrity) requirements of business processes can be modeled.

The responsibility modeling approach was first proposed in the third generation (Dobson, 1990) based on the idea that finding and modeling work responsibilities as the key point of departure for securing system. This idea was later adopted by Backhouse & Dhillon (1996) and McDermott & Fox (1999) using a socio-technical lens rather than a modeling approach. Applicability to SIS design is a major concern when it comes to responsibility modeling since there is no model that explains how responsibilities can be integrated into IS design process. The viable approach of Hutchinson and Warren (2000) is incorporated into the third generation domain due to the technical organizational role attributed to IS security. Limitations of third generation approach are discussed in detail in literature review section.

The emerging “socio-technical” generation takes the organizational security requirements as a first step in developing SIS design and does not just replace the organization's unique information security requirements for a generic list of predefined protection means suggested by some information security experts. The socio-technical generation acknowledges the interaction between people and technology. It highlights the interrelatedness of social and technical aspects of an organization as a whole. Securing the technical aspects of an organization cannot be achieved without considering social aspects of processes. This generation includes soft system approach by James (1996), responsibility modeling (Dhillon, 1997), and survivable systems (Karyda et al., 2001).

The responsibility modeling SIS design approach believe that the security requirements can be modeled by exploring the role responsibilities in organizations Dhillon (1997). Karyda et al., (2001) argued that instead of focusing on secure IS, research should focus on viable IS. In other words, a secure system should survive or be survivable in the face of attacks. The “Orion strategy” proposed by James (1996) stems from Checklands (1981) soft system methodology. James (1996) was the first to embed user participation in designing secure IS. The Orion strategy emphasizes participation of people in defining what needs to be done in terms of managing security in information systems.

Emergent literature focuses on socio-technical approaches to IS design, perhaps the biggest problem with the existing socio-technical design paradigm is developmental duality, i.e., they do not provide any guidance as to how their design approaches could be integrated into the IS development process (Baskerville, 1992). Applicability to information system and software development (Baskerville, 1992) is needed to explore whether the secure information system design approach can be integrated into information systems. The importance of the research problem stemmed from the need to have an applicable socio-technical approach for secure IS design that includes security requirements from both aspects (social and technical).

1.4. Definitions of Terms

Three classes of definitions are required to establish the conceptual boundaries of this research. For the purposes of this dissertation, definitions of select terms used are provided.

Information System is defined as an aggregate of information handling activities at a technical, formal and informal level of an organization (Liebenau & Backhouse, 1990). This is consistent with Karyda et al., (2001) view of an information system as "*a human activity system comprising five elements, namely hardware, software, data, procedures and, above all, people…in order to support human activities in the context of an organization.*" (Karyda et al., 2001 p. 454.). The hardware, software, and data comprise the technical level. Procedures comprise the formal level. The formal level is bounded by rules used to control organizations. This research proposes to model the context of procedures, behaviors and human activities and define the context of procedures in terms of rules in the formal level to control organization. Finally, people or the informal character of information systems is expressed in people's assumptions, expectations, beliefs and informal goals, which comprise the informal level of an information system.

Information System Security has been traditionally defined by computer science researchers from a technical perspective adopting the CIA triad (Bishop, 2003). This definition does not align with the view of IS security adopted in this research. The organizational role in information security is socio-technical. The definition adopted in this dissertation considered information systems security as a well-informed sense of assurance that information risks and controls are in balance (Anderson, 2003). This definition aligns well with this research as it acknowledges the existence of formal levels in information systems in terms of organizational control.

Business Process and Business Process Modeling (BPM) have many definitions as discussed in the literature review. This research followed the definition by D'Aubeterre et al., (2008a) which views a business process as a set of coordinated activities, enacted by human or software agents that exchange knowledge resources to achieve business objectives.

1.5. Summary

This research investigated secure IS design. The research problem was introduced in chapter 1. Discussion of relevant literature on secure IS design approaches is presented in chapter 2. The research design and methodology are presented in chapter 3. Action research is presented in chapter 4. Findings, implications, and limitations of this study are presented in chapter 5. Conclusions are presented in chapter 6.

Chapter 2

Literature Review

2.1. Introduction

In the first section, the evolution of IS development methodologies in terms of generations and paradigms is presented (Hirschheim & Klein, 1992). Generations and paradigms in SIS design are then presented (Siponen, 2002). An explanation for positioning this research in socio-technical paradigm is then presented. This review provided a basis for the suggested approach based on organization based access control (OrBAC) (Kalam et al., 2003) as the kernel theory.

2.2. IS Design Approaches

IS development (ISD) methodologies as portrayed by (Hirschheim & Klein, 1992) are presented first. Chronologically, there are seven generations of ISD approaches (Formal, Structured, Prototyping, Sense-making, Trade-union, Emancipatory and Socio-technical) approaches. These methodologies fall within four paradigms (functionalism, social relativism, radical structuralism, neo-humanism) based on their ontological and epistemological assumptions (Hirschheim & Klein, 1992).

2.2.1. Formal life cycle approach (classical)

The earliest of ISD generations was inspired by system development approach in software engineering. This generation was known as the formal, traditional or classical approach. The term software engineering started in the early 70's, which referred to well-structured methods and tools for program design, implementation, and testing once system requirements are given.

Too many projects were failures, the importance of information systems planning for determination of requirements and priorities of projects was noted in IS management. Blumenthal (1969) suggested “planned evolution” as a methodology for orderly, organization-wide IS development based on his experience with systems development. IS development depended on codified techniques that eventually were formalized into a “methodology”.

IS are built from the requirements elicited by the system analyst from the users. User requirements elicitation was considered difficult but not a noncontroversial exercise. Users provided information is the basis for IS requirements. This rational strategy formed what is called a “System Development Life Cycle” (SDLC). SDLC is the “classical” ISD approach which iterates through five general steps (requirements analysis, design, implementation, testing, and evolution). According to Elliot (2004), SDLC "*originated in the 1960s, to develop large-scale functional business systems in an age of large-scale business conglomerates. Information systems activities revolved around heavy data processing and number crunching routines*".

2.2.2. Structured approaches

The formal SDLC approach suffered from major problems that led to the development of the structured approach (Hirschheim & Klein, 1992). From a system developer perspective, the design of the system is almost impossible because of changing user requirements. From the users' perspective, it was hard to know in advance what the implemented system should look like. To address these issues, a more structured approach was suggested using a "sign off" and the "structured walk through" (Hirschheim & Klein, 1992).

The sign off allows the system analyst to work to an agreed specification that was signed off by the users. The structured work through allows the users to have an understanding of what the finished product would look like as they were formally "walked through" the details of the system design during formal sessions. IS development was still perceived as a technical process, although the social consequences were acknowledged and somewhat considered in the structured walkthrough.

2.2.3. Prototyping approaches

In the late 80's, the structured and formal approaches became problematic as they slowed down the design and development of systems. Users could no longer wait for a formal life cycle to finish, the implemented system would no longer meet the user requirements after two or three years of development. The idea emerged that users needed to experience the application software to better understand what the application would look like. Naumann & Jenkins (1982) and Earl (1978) suggested using a prototyping approach. A prototype is a scaled-down variant of the final system that exhibits some of its salient features to allow the users to get a feel of the interface and the computational power.

2.2.4. Sense-making approaches

One concern behind the structured approaches was problem formulation. Some argued that it can be easily tackled using a scientific approach to problem-solving (Newell & Simon, 1972). Checkland (1981) had a different perspective with his soft system methodology (SSM). Checkland (1981) believed that there must be a richer interpretation of the "problems of problem formulation". "Problems" or user requirements are not easily articulated, it may be misleading to assume that a problem exists rather that one is constructed between various stakeholders (Hirschheim & Klein, 1992).

Following this approach and as an extension of SSM, "sense making" methodologies emerged. The argument behind this approach is the need to develop approaches that would lead to a better mutual understanding between users and developers to avoid problem formulation. "sense making" can be defined as "*the modes in which a group interacts to interpret their environment and arrive at socially shared meanings*" (Klein and Hirschheim, 1987, p.302). System development in "sense making" is perceived as a social process. However, the projects that surfaced from sense making approach never matured into a complete methodology (Klein and Hirschheim, 1987).

2.2.5. Trade Union approaches

In trade union approaches, system development is seen as part art, part science, and part class politics; it's conceived as a very much social process rather than a technical one. It was a reaction to the negative social effects of generation 1 and 2 approaches. The approach argues that control of system development should be in the hands of the workforce rather than management (Kubicek, 1983). Proponents of this approach felt that a socio-technical approach is a form of deceit to reduced worker resistance to systems that served mostly the interest of managers and offered little to improve worker position. A set of projects immersed that introduced a set of general guidelines, tools, and techniques that would allow the trade union to dictate the system direction (Hirschheim & Klein, 1992). However, there is not much research on evaluation of this approach.

2.2.6. Emancipatory approaches

An emancipatory approach is merely a conceptual approach supported by the emancipatory ideals. There are no developed projects that adopted the emancipatory theme. The emancipatory approach conceives system development as a social process and sees the need for sense-making (called "mutual understanding" in this approach). The difference from other approaches is in the orientation towards emancipation through rational and emancipatory discourse.

The goal of systems development is a system that would not only support emancipatory discourse but also a mutual understanding of all its users. Hershihem & Klien (1994) explored the emancipatory principles and discussed how they might be applied in information systems development. Existing ISD methodologies only partially embrace emancipatory principles (Hershihem & Klien 1994).

2.2.7. Socio-technical and participative approaches

System development in this approach is seen as part art and part science. It is viewed jointly as a social and technical process. The structured and prototyping approaches progressed the field greatly, but there were few issues identified by IS research (Mumford (1998); Land & Hirschheim (1983)). IS researchers felt that sign offs and structured walk-throughs were potentially helpful but were fundamentally misguided in their ability to elicit true user involvement (Hirschheim & Klein, 1992). System development had traditionally focused on the technical systems rather than the social systems. This led to information systems that might have been technically sophisticated but was not ideal from a social or work standpoint (Hirschheim & Klein, 1992).

The socio-technical approach adopted in this study suggest that interventions should lead to an improved social as well as technical systems. System development in the socio-technical approach uses system development as a vehicle to rethink the social work environment in which the new system would be implemented. According to (Hirschheim & Klein, 1992) all socio-technical approaches should focus on having the users not only be involved in systems development but take control of the process; and having systems development to be used to redesign the work situation leading to an improve social and technical system.

2.3. Business Process and Business Process Realization

D'Aubeterre et al., (2008) argued that SIS design must incorporate security as a functional requirement in the early stages of requirement specification and analysis of IS. To do so, there must be an understanding of how business processes are coordinated to achieve their objective. This allows eliciting security requirements from the context of business processes. A business process is a set of coordinated activities, enacted by human or software agents that exchange knowledge resources to achieve business objectives (D'Aubeterre et al., 2008). An organization's business processes comprise value activities that create customer value (Porter & Millar, 1985). These activities must be coordinated to achieve the business goals effectively. Business processes require coordination mechanisms to manage the interdependencies of their constituent activities (Malone, 1987).

The security of knowledge resources must be ensured. Information and knowledge resources sharing may lead to possible unauthorized access and usage which in turn would cause loss of assets. An organization may incur significant costs without appropriate and timely authorization access to information resources when performing business activities (D'Aubeterre et al., 2008).

From an IS perspective, IS researchers are moving away from the security technical viewpoint toward a socio-organizational perspective (D'Aubeterre et al., 2008). This movement may lead to more holistic IS security research where organizational security aspects are incorporated in the design and development of SIS. Hence, capturing the business process context will allow integrating security in the design and development of SIS.

2.4. SIS Design Approaches

Secure IS design was inspired by many disciplines including IS development and computer science approaches. Baskerville (1992) provided an overview of conventional SIS design approaches (normative standards, risk management, formal methods, and common sense principles). More recently, Siponen (2002) conducted an analysis of contemporary SIS design approaches (see Appendix A.). Some of these methods either lack serious research or suffer from development duality. These methods cannot be easily integrated to IS development process (Siponen, 2002).

2.4.1. Traditional SIS design (First and Second generations)

The first and second generation (Naturalistic-Mechanistic) of SIS design were focused on what can be done as opposed to what needs to be done from a technical point of view. This generation was described in Baskerville (1992) which included normative standards, risk management, formal methods, and common sense principles. Normative standards referred to checklists, management and maturity standards. The idea behind checklists is to write down what are the possible security solutions that can be identified and turn it into a list.

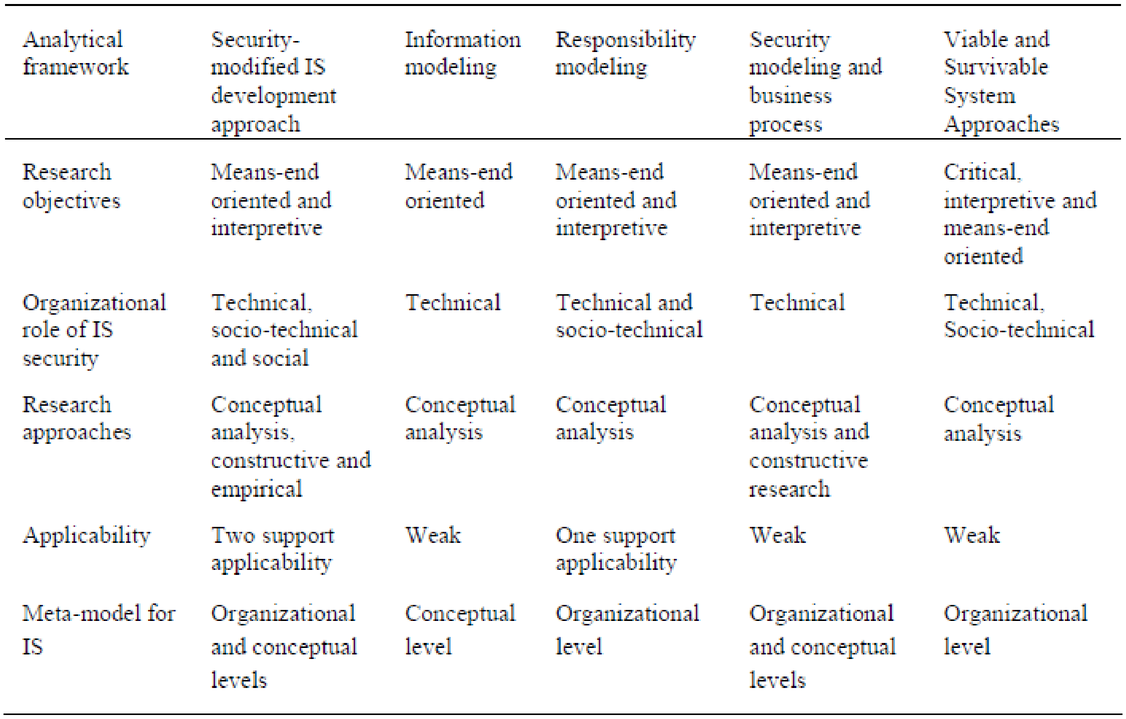
Risk Management (RM) can be described as an interpretive means-oriented technical communication tool to help manager understand risks. Formal methods suggest that IS or software development should be carried out by formal methods. These formal methods are supported by the use of logic and mathematical modeling. However, both RM and formal methods suffer from development duality. Common sense principles are developed on the basis of practical experience and neglecting related work.

2.4.2. Contemporary SIS design (Third and Fourth generations)

Modern SIS design approaches consist of the third generation (IS modeling approaches) and the fourth generation (socio-technical approaches) (see Appendix A.). Siponen organized contemporary SIS design approaches based on an analytical framework of research objective, organizational role of IS security, research approach, applicability, and the meta-model for IS design process. In consensus with Baskerville (1992), Dhillon & Backhouse (2001) and D'Aubeterre et al., (2008), Siponen (2002) believes that the organizational role of IS security (see Table 1) should be socio-technical.

Researchers call for alternative SIS design approaches that embrace the social and socio-technical view. Third generational approaches focused on IS modeling, and fourth generational methods emphasize socio-technical design. This explains why contemporary approaches are classified under two different generations, as they differ in their underlying assumption of the organizational role of IS security. Virtual methodology (Hitchings, 1996) and the Orion strategy (James, 1996) are aligned within the fourth generation (socio-technical) while other approaches are aligned within the third generation (IS modeling).

Table 1. An Overview of Contemporary SIS design approaches (adopted from Siponen (2002))



Analytical framework	Security-modified IS development approach	Information modeling	Responsibility modeling	Security modeling and business process	Viable and Survivable System Approaches
Research objectives	Means-end oriented and interpretive	Means-end oriented	Means-end oriented and interpretive	Means-end oriented and interpretive	Critical, interpretive and means-end oriented
Organizational role of IS security	Technical, socio-technical and social	Technical	Technical and socio-technical	Technical	Technical, Socio-technical
Research approaches	Conceptual analysis, constructive and empirical	Conceptual analysis	Conceptual analysis	Conceptual analysis and constructive research	Conceptual analysis
Applicability	Two support applicability	Weak	One support applicability	Weak	Weak
Meta-model for IS	Organizational and conceptual levels	Conceptual level	Organizational level	Organizational and conceptual levels	Organizational level

The Information modeling approach is considered in the “IS modeling” generation. This approach is motivated by the desire to build security notations (Hirshheim et al., 1995), specifically for developing secure databases. The organizational role of IS security is purely technical. However, this approach is hard to integrate into IS development since this approach adopts a low-level (technical) database development oriented view (Siponen, 2002).

The security modeling and business process approach was suggested by a group of German researchers (Röhm et al., 1998). They suggest a framework where confidentiality and integrity requirements of a business process can be modeled. The framework consists of a three-layered architecture for business process security. In the third layer, the high-level security requirements of business processes are graphically analyzed. In the second layer, these are translated into more a formal, intermediate language, and the security elements are identified and divided into security blocks. This framework is mainly concerned with layer three. The third layer provides five perspectives.

The perspectives are informational, functional, dynamic, organizational and business process perspective. The perspectives are created using entity relationship diagrams, state transition diagrams, and data flow diagrams to model information and business aspects (Pernul et al., (1992); Hermman & Permul (1999)). They argued that information modeling is needed to communicate the security requirements between the different people involved in the development in question (Herrman & Pernul 1998 p. 766). While this approach may succumb the problems of development duality. Researchers did not explain how this approach can be integrated into IS development. In addition, the focus is on technical systems so the organizational role of IS security is purely technical. This, in turn, ignores socio-technical and participative aspects of IS design. While dissemination of information is discussed in the design paradigm, eliciting security requirements from stakeholders is not addressed.

The responsibility modeling approach argued that security requirements are found by understanding the role responsibilities in organizations. Backhouse & Dhillon (1996) and (Dhillon, 1997) among other researchers (McDermott & Fox, 1999) argued for using responsibility as a basis for IS security development. A major problem with this approach is development duality with the exception of McDermott & Fox (1999) which is clearly applicable to IS development approach. McDemott & Fox (1999) view varies from technical to socio-technical in terms of the organizational role of IS security.

Viable and survivable system approaches (Hutchinson & Warren (2000); Karyda et al., (2001)) are theoretically grounded on Beers viable system model (Beer, 1984). The model consists of five systematic functions, which need to be performed in order for an organization to be viable (or survivable). These systematic functions are:

1- System One: the 'operational elements' that produce the system and interact with the external environment. These elements are themselves viable systems.

2- System Two: the 'co-ordination' functions that ensure that the operational elements work harmoniously.

3- System Three: the 'control' activities, which maintain and allocate resources to the operational elements.

4- System Four: the 'intelligence' functions that consider the system as a whole -its strategic opportunities, threats, and future direction. They also interface with the environment.

5- System Five: the 'identity' function, which identifies self-awareness in the system.

However, the SIS design approach taken by Krayda et al., (2001) is quite radical; shifting focus from secure system to what is called viable or survivable IS. "*We propose a new framework for building secure information systems, or as we suggest them to be called, viable information systems*" (Karyda et al., 2001 p. 453). The organization role of IS security is socio-technical. Albeit, Karyda et al., (2001) does not explain how this approach can be applied to IS design process, it is a possible venue for research."security-modified IS development approach" was used to describe approaches that are influenced by IS or software development methods.

The theoretical lens varies in these approaches as they belong to different generations. Approaches from the third generation (IS modeling) are logical modeling (Baskerville, 1989), Spiral approach (Booysen & Eloff, 1995) and IS security planning methodology (Straub & Welke, 1998). Approaches from the fourth generation (socio-technical) include the Orion strategy approach proposed by James (1996), and virtual Methodology approach proposed by Hitchings (1995).

Focusing on what needs to be done and using a security modified IS development approach, two SIS development approaches can be integrated into IS development process (James, 1996; Hitching, 1995). James (1996) introduced Orion strategy, which suggests that security problems and potential problem situations existing within business organizations should be in effect, handed back to the people who use, and are responsible for, those systems (James, 1996). James (1996) was perhaps the first to embed user participation in the designing secure IS. James (1996) was closer to the social view on the organizational role of IS security following Checklands (1981) soft system methodology.

This research took a similar approach to James (1996) and Hitchings (1995) by considering the socio-technical role but with a few minor differences. However, the focus is on eliciting requirements in terms of business process contexts and capturing security requirements. This allows minimizing user involvement unlike James (1996) which is considered to be more radical towards the social view. The novelty in the approach in this study is that it described how contextual, emergent and changing requirements can be incorporated into the design of IS meeting all meta-design requirements as defined by Siponen et al., (2006). The approach explained not only what needs to be done, but also how it can be done.

2.5. Summary

Literature review on the evolution of secure information system has been provided since 70's following Baskerville (1999) and Siponen (2002). Design approaches have matured to meet the need for modern information systems. Based on the review, there seems to be a consensus that the role of IS security must be socio-technical. Secure information system design with socio-technical aspects in mind is the goal of this study. The conceptual modeling of context has been used in different areas of research (Myers, 1997) e.g. modeling reasoning and expert systems (Brézillon, 2015). Context graphs have also been used in modeling tasks and practices (Dichev, 2001). The use of context graphs has also been suggested to model experiences Batarseh et al., (2013).

Chapter 3

Research Design and Methodology

3.1. Introduction

This research followed design science research paradigm in IS to develop a method for SIS design (March & Smith, 1995). A distinction between design science and design science research is made here. Design science is research into or about design; and design science research, which this research follows, is research using design as a research method or technique (Vaishnavi & Kuechler, 2004). A brief introduction to design science and design science research in IS is presented first. Design science research methodology process model (Perffers et al., 2007) is then followed.

Design science research is motivated by the need to improve the environment by introducing new creative and innovative artifacts (Simon, 1996). The design science paradigm originated from engineering and sciences of the artificial (Simon, 1996). Design science in IS research creates and evaluates information technology artifacts intended to solve identified organizational problems. The artifacts should demonstrate a contribution to knowledge (Gregor & Hevner, 2013).

The application domain for design science includes people, organizations, and technical systems (Hevner, 2007). The result of design science research in IS is a purposeful information technology artifact created to address an important organizational problem.

The artifact must be described effectively, enabling its implementation and application in an appropriate domain (Hevner et al., 2004).

Design science research involves the creation of new knowledge through design of novel or innovative artifact and analysis of the use and/or performance of such artifact along with reflection and abstraction to improve and understand the behavior of aspects of information systems (Vaishnavi & Kuechler, 2004). Design science research in IS is conducted within design processes and produces four types of design artifacts (March & Smith, 1995). The artifacts could be methods, models, constructs, and instantiations. The artifacts are designed to address unsolved problems. They are evaluated with respect to the utility provided in solving those problems (Hevner et al., 2004). The overall design science research process model includes six steps: problem identification and motivation, definition of the objectives for a solution, design and development, demonstration, evaluation, and communication (Peffers et al., 2007). These steps are discussed in the next subsections.

3.2. Problem Identification and Motivation

Hevner et al., (2004) argued that IS research is conducted in two complementary balanced phases, a behavioral science paradigm and a design science paradigm. The goal of the behavioral paradigm is to provide Truth (justified theory or true statements corresponding with real world phenomena) while the design science paradigm is focused on Utility (building artifacts that are effective).

The behavioral science research paradigm develops theories and justifications to explain or predict a phenomenon related to business needs. The driving business need in this study is securing IS from starting with early design stages. Practitioners who are managing, designing and implementing IS fail to secure the business interactions and operation of people within organizations and information technology. Following the design science research paradigm, this research built and evaluated an artifact designed to meet this driving business need and provided applicable knowledge to secure business processes.

3.3. Definition of the Objectives for a Solution

The objective was to provide a secure information system design that meets well-defined criteria (Siponen et al., 2006). The solution was presented as an artifact using design science. Design science artifacts in IS are theoretically grounded on a design theory or “kernel” theory (Markus et al., 2002). Design theories are prescriptive theories, based on scientific theory, technical information, and imagination that explain how the process of designing an information system can be feasibly and effectively carried out for a particular set of requirements (Walls et al., 1992). Design theories are considered as practical knowledge used to support design activities (Goldkuhl, 2004). This research introduced a nascent design theory as per (Gregor & Hevner, 2013) to address the identified problem discussed in the previous section.

A secure design must capture socio-technical aspects of IS, it must also understand the context of organizational processes and activities. These contexts are captured in the suggested artifact using Organization Based Access Control (OrBAC) context model (Cuppens & Miège, 2003). The next section contrasts Role-Based Access Control (RBAC) in IS with OrBAC and describes how OrBAC is used in the design artifact.

Confidentiality is often enforced using a form of authorization known as access control. Many theoretical access control models exist. The discretionary access control did not adopt well to IS in organizations (Ferraiolo et al., 1995). The introduction of databases required users to own information to run their operations and activities. The security of business processes in organizations became a burden. The role-based access control (RBAC) access control model (see Figure 1) was one of the early models that introduced the concept of "Role" (Ferraiolo et al., 1995). The design of RBAC states that a user can fulfill a role, and therefore that user can own the information related to that role. The security design aspect of IS relies predominantly upon security specialists towards building a dedicated group of IS roles using technical methods. Security specialist will create user roles (system wide access control LDAP in windows) and\or application specific roles (payroll system within organizational IS).

RBAC strictly focuses on roles. The roles are statically defined by the experts (e.g. user, admin) and do not follow organizational roles and context. During IS design stages, these roles are what define security controls that need to be placed in terms of authorizations and not requirements as defined by IS users.

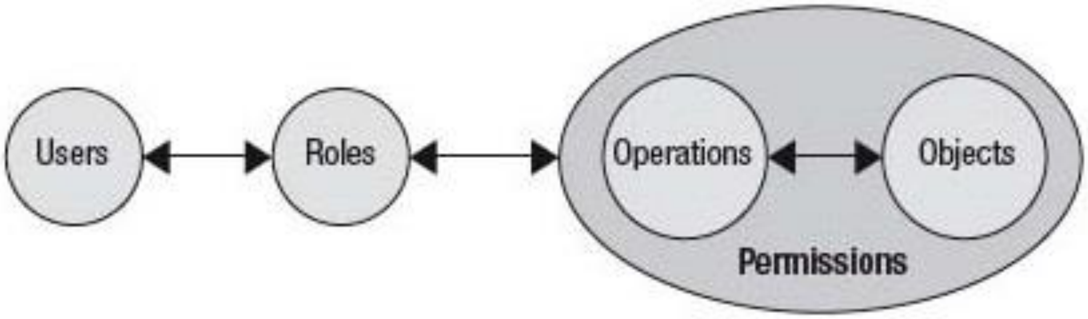


Figure 1. Basic RBAC as in NIST Standard 359-2004

There are limitations when adopting RBAC in terms of access control methodology in IS design. These limitations are listed along with limitations found in (Kalam et al., 2003) that inhibit dynamic access control in Table 2. These RBAC limitations are contributing to SIS failure in capturing the nature of processes in IS, which in turn creates room for vulnerable IS. Control of IS access must consider the custom nature of organizations and clear the confusion between actions and roles.

Table 2. RBAC Limitations in IS

No.	RBAC Limitation
1	The concept of permission is primitive. When specifying the security policy of a given application, the RBAC model must be refined to make explicit the structure of permissions.
2	The concept of role hierarchy is not free of ambiguity. In particular, it is generally incorrect to consider that it corresponds to an organizational hierarchy.
3	The ability to specify a permission that depends on a given context. More precisely, if a given permission is granted to a given role, then all users that play this role will inherit the given permission.
4	RBAC only enables the administrator to specify permissions.
5	Additional limits appear when the RBAC model is used to specify the security policy of a system that includes several organizations.
6	RBAC does not incorporate the content and context of the information workflow and does not separate task (action) from role.

The existing SIS design methods merely define general strategies and principle functions for IS design and do not explicitly state how these strategies can be integrated into IS development as previously mentioned. They also do not explain the dynamics of access control in such IS. The objective of this research was to create a design method that would fully explain the security design aspects in all SIS development stages, from requirement analysis to design and implementation stages.

Surely enough the design theory behind the proposed SIS design approach requires both, a unique type of IT solution (IT product design) and a unique design strategy (IS development approach) to satisfy unique “meta-requirements” (Markus et al., 2002). Notably, meta-requirements for SIS design have been proposed by Siponen et al., (2006). In the next section, the artifact that incorporates context from an IS perspective into SIS design model is presented. Meta-notations to represent analyzed security requirements in the design of IS are presented.

3.4. Design and Development

OrBAC constructs are considered when defining secure IS requirements in the proposed design method. The OrBAC model was originally proposed in the literature as an alternative to existing access control models to specifically address the problem of specifying contextualized and emergent technical security rules (Kalam et al., 2003). Unlike other access control models, OrBAC can apply rules that specify contextual permissions for specific circumstances. An example would be an IT technician gaining administrator privileges to a core service to introduce a hotfix when the actual administrator is not available. Alternatively, in health care, a physician having special permissions in the context of urgency. In addition, from a control perspective, the OrBAC model can specify granular authorization including prohibitions, obligations, and recommendations; unlike classical access control models which are only restricted to permissions.

Cuppens and Miege (2003) define an organization as an organized group of active entities, i.e. subjects, playing some role or other. A subject plays a role in the organization corresponding to some agreement between the subjects to form an organization. Kalam et al., (2003) used organizations as an abstraction layer between subjects and roles, objects, and views, action, and activities. The definition of the context also depends on the organization. Contexts are used to specify the concrete circumstances for organizations and are only defined by concrete entities. The entities, organization, subject, object, action, and context are linked together by the relationship Define (see Appendix C).

OrBAC is a contextualized access control method that addresses the limitations of RBAC. Conceptualizing access control in processes using OrBAC provides the comprehensiveness required and adaptation to IS security requirements. Adoption of context from the OrBAC model allows modeling the complexity of socio-technical IS or the human activity system. OrBAC constructs are considered when defining secure IS requirements in the design method. The objective of OrBAC is to specify the access control policy at the organizational level that is independent (abstract) of the implementation of this policy.

The abstract level defines the concepts of organization, role, activity, view, context and security rules to express the abstract policy. The abstract policy, specified at the organizational level, is specified-using roles, activities, and views, which respectively abstract the concrete subject, actions, and objects. The role of a subject is simply called a role as in the RBAC model.

The organization employs a subject so it defines that role. On the other hand, an action is an abstraction of activity. Different organizations may consider activities to belong to different actions. The abstraction of an object is called a view. A view is an organizational concept used to structure the policy specification for using objects, i.e. a view groups objects on which the same security rules apply.

In OrBAC, one can define that a subject may have the permission, prohibition or obligation to do an activity on some object given an associated context is true (see Figure 2). The formalism of the OrBAC model uses first-order logic notations which allow each organization to specify its own security rules. These security rules are represented using an array of first order predicates. Notations and constructs from the OrBAC model are adopted in the design method. Even though OrBAC constructs are adopted, the unique contribution of this work is an artifact that describes how an IS can be securely designed using these simple constructs:

- Organizations (org): a central entity of OrBAC may represent multiple organizations or an organized group of subjects.
- Subjects (s): person, actor, entity or an automated agent.

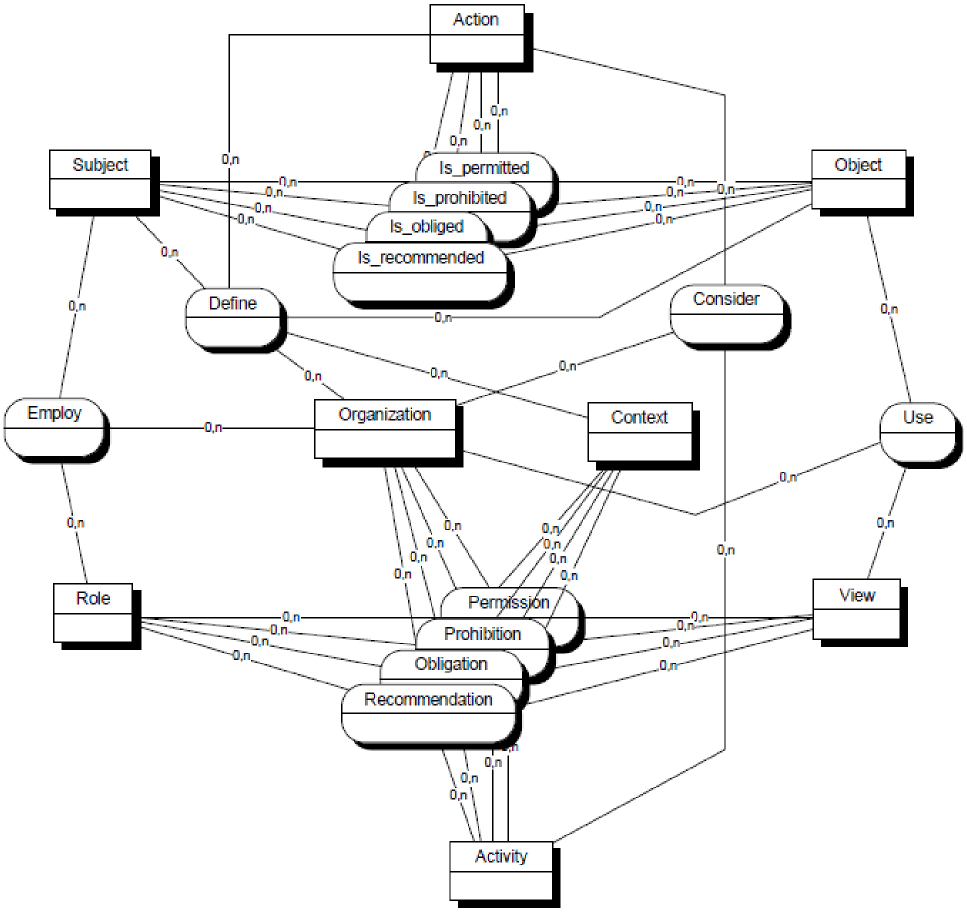


Figure 2. OrBAC model (El Kalam et al., 2003)

- Actions (a): means for subjects to access objects.
- Objects (o): static entities e.g. files, records.
- Roles (r): a set of subjects to which the same security rule applies.
- Activities (alpha): a set of actions to which the same security rule applies.
- Views (v): a set of objects to which the same security rule applies.
- Contexts (c): a condition in which rules only apply when the condition is true

The context concept has been introduced in the OrBAC model in order to express dynamic rules (Cuppens & Miège, 2003). Contexts correspond to any constraint that joins a subject, an action, and an object. Contexts may be temporal, spatial, user-declared, prerequisite or provisional. Cuppens & Miege (2003) defined the following contexts:

- Temporal (depends on the time at which the subject is requesting an access to the system)
- Spatial (depends on the subject location)
- User-declared (depends on the subject objective)
- Prerequisite (depends on characteristics that join the subject, the action, and the object)
- Provisional (depends on previous actions the subject has performed in the system).

The CSIS artifact suggests adopting context from the OrBAC model using a socio-technical lens to incorporate security during requirement analysis phase to ensure secure business processes. A business process context is defined from people's point of view. In the CSIS artifact, context must be with the same level of abstraction as the organization. Contexts must validate roles, activities, and views. This way the required level of abstraction from explicit implementations of actions can be provided as actions are separate from roles.

As with current literature, this research ontology on security is described as a functional requirement in the analysis and modeling of business processes. This helps achieve the research goal in incorporating security analysis and design into the overall IS analysis and design methodology. The conceptualization used in the CSIS artifact is consistent with OrBAC access control model from an IS point of view.

OrBAC constructs are used to define security IS requirements in the suggested design method. The nascent design theory (Gregor & Hevner, 2013) in this conceptualization is that the underlying process activity implementation i.e. interaction between subjects, objects and actions is completely abstracted. This allows the method schematic to be drawn in an abstract, holistic and systematic manner. The following rules represent the modeling concepts for secure business processes during requirement analysis.

- Organizations define the abstract Roles, Activities, and Views:
    - A subject fulfills an organizational role. Organizational roles are fulfilled by subjects.
    - An activity is considered a set of actions; actions are a simple operation that collectively represents an activity.
    - A view uses a set of objects. objects are used by a view.
- A role may have the permission, prohibition, recommendation or obligation to do an activity coordinated on some view if and only if the given context is true.

Using these rules, an artifact to describe how an IS can be designed with security in mind can be deduced. This artifact is considered a design artifact for Contextual SIS (CSIS) design (see Figure 3). The artifact presented in Figure 3 will allow IS practitioners to embed IS security requirements in all IS design and development stages, from requirement analysis, design and through the implementation of IS.

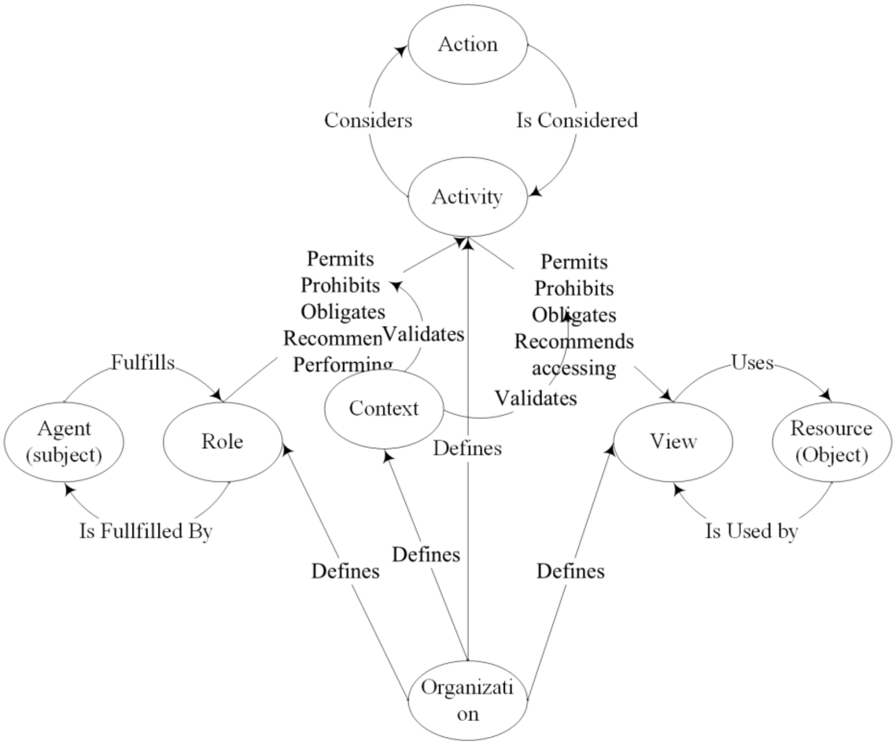


Figure 3. CSIS Design Artifact

An organization aims to achieve a business objective and defines a set of general roles to achieve these objectives. An agent assigned by the organization fulfills these roles. The agent is restricted by the activities allocated to that role. These activities are the business processes required to achieve the business objective. Each activity is restricted by specific actions and the views allocated to them.

The view is specific to certain objects and is only granted based on the allocated activity. IS practitioners are responsible for defining these allocations based on business and IS requirements. These requirements are also used to extract the business process context. The defined context is used to check whether the role is obligated, prohibited, obligated or recommended to perform the given activity on the given view.

The changing and shifting organizational requirements can be accounted for using organizational context (Siponen et al., 2006). The developed artifact uses an activity context as a core control to IS security. After the implementation of the IS, contexts can be tweaked to account for those changes. A change in the nature of activity conducted in IS reflects back to the context that validates the activity to ensure the security of said activity and view.

The artifact can be driven to later stages. Requirements analysis stage can define the organizational roles, views, and activities. The context of these activities can be defined during design stages. The actual entities subjects, objects, and actions are later defined during the implementation of IS. However, a strict review process on context change must be done by security experts to ensure the emergent contexts based on business needs do not affect the state of information security assurance in IS.

Using context, different sets of high-level activities and scenarios can be accounted. IS resources (objects) should only be available to agents (subjects) when the agent's associated action falls under his/her role and that role is known to have activities that require viewing these resources (objects). This form of control allows for much more agility in terms of adaptation to requirements based on organization needs and at the same time be able to easily find access anomalies based on context nature.

Situation awareness theory (Endsley, 1995) is used to measure IS practitioners awareness of security policies, regulations, and constraints generated by the CSIS artifact.
Endsly (1995) defines situation awareness as "*the ability to perceive the status, attributes, and dynamics of relevant elements in the environment*". Situation awareness is
the perception of environmental elements with respect to time or space, the comprehension of their meaning, and the projection of their status after some variable has changed, such as time, or some other variable, such as a predetermined event. It involves being aware of what is happening in the vicinity, in order to understand how information, events, and one's own actions will affect goals and objectives.

D'Aubeterre et al., (2008) extended situation awareness theory to define security awareness, which "*assess the comprehension of security policies and constraints in the analysis of security requirements*". The utility of the CSIS artifact is assessed by assessing the security awareness generated by the CSIS artifact against the best known SIS design methods, namely, secure activity resource coordination (SARC) conceptualization of secure business process (D'Aubeterre et al., 2008), security enriched use cases (Siponen et al., 2006) in capturing the security requirements and security aspects of IS. The research model to test security awareness generated by these SIS design methods is presented in Figure 4. Hypotheses are discussed in the next subsection.

It is evident that the CSIS artifact should be informationally equivalent (Larkin & Simon, 1987) to other SIS design methods. CSIS must ensure that it completely captured IS and business process requirements with no information loss. This describes the first research hypothesis.

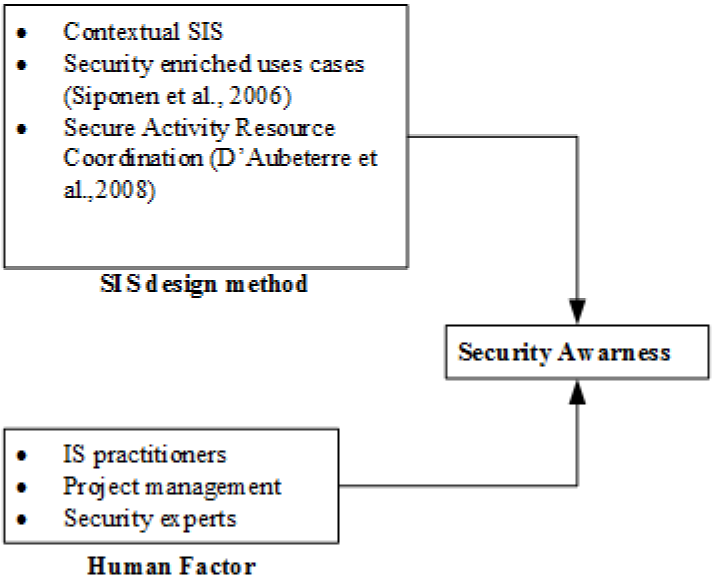

Contextual SIS
Security enriched uses cases (Siponen et al., 2006)
Secure Activity Resource Coordination (D'Aubeterre et al.,2008)
SIS design method
Security Awarness
IS practitioners
Project management
Security experts
Human Factor

Figure 4. SIS Design Method Research Model to Test Security Awareness

H1: SIS design developed using CSIS artifact is informationally equivalent to SIS design developed using security enriched use cases or SARC.

CSIS artifact enabled the use of multiple contexts e.g. temporal, spatial, etc., the design of IS will be context-aware (Abowd et al., 1999). CSIS artifact must generate higher security awareness of security requirements of IS than other SIS design methods. IS practitioners and security experts should develop a higher level of security awareness when using CSIS as opposed to other SIS design methods.

H2: SIS design developed using CSIS artifact will generate higher security awareness than SIS developed using security enriched use cases or SARC.

The individuals' characteristics influence the perception and interpretation of reality (D'Aubeterre et al., 2008). These characteristics influence the user's comfort with reading diagrams (Allen & March, 2006). This research tested if users with experience in IS analysis and security would develop a higher level of security awareness using CSIS than individuals without experience.

H3: SIS design developed using CSIS artifact will generate higher security awareness than SIS developed using security enriched use cases or SARC for experienced IS practitioners and security experts.

The next section demonstrates how the CSIS design artifact can be used in the design of SIS. The demonstration provides evidence for supporting the first hypothesis. A use case scenario is described in the following section.

3.5. Demonstration

The artifact must show evidence that it is useful. During demonstration stage, the artifact is used to solve one or more instances of the problem using knowledge of how to use the artifact to solve the problem (Peffers et al., 2007). The utility of the CSIS design artifact is in providing awareness and knowledge about functional security requirements to business analysts in order to systematically translate complex business processes into a secure design. Here, a demonstrative example is provided in Table 3. In the next section, field-testing and rigorous evaluation in the application domain is discussed. The example is adopted from IS literature (Jaaksi, 1998; Siponen, 2002).

Table 3. Booking Use Case (Jaaksi, 1998; Siponen, 2002)

Use Case:	Booking
Version:	1.0
Functional Summary:	Preconditions: Booking and customer databases exist. Exceptions: If information on certain journey is not available an appropriate error message is produced
Frequency:	Several times a day
Usability requirements:	A booking clerk books journeys for customers. Any database query and booking must be able to complete in less than 30 seconds
Actor:	A clerk

The following are the conceptual modeling rules:

- Booking organization defines the abstract Booking Roles, Activities and Views:
  - A booking agent fulfills a booking role. Agents fulfill booking roles.
  - A booking activity is considered a set of booking database queries; booking database queries are considered booking activities.

- A booking view uses a set of databases, booking, and customers. Booking and customers databases are used by the booking view.

- Any booking agent fulfilling the booking role may have the permission, prohibition, recommendation or obligation to run booking activities coordinated on booking view if and only if the given booking process context is true. A booking agent (see Figure 5) who fulfills the booking role books journeys for customers, several times a day. A temporal context may validate a period where the agent is requesting access to the system. A spatial context may validate whether the agent is at the booking agency during the process. Other contexts can be user declared based on the Booking Co. organization needs Figure 5.

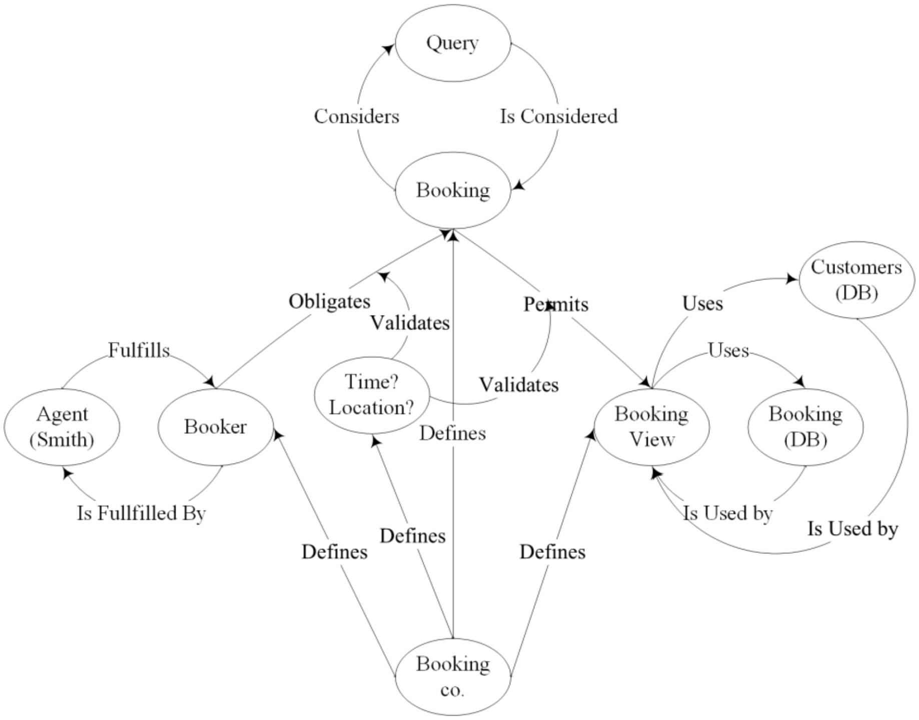


Figure 5. Booking Co. Secure Business Process Realization

The demonstrative scenario shows evidence for supporting the first research hypothesis. The resulting artifact encapsulates information captured in a use case scenario. SIS design developed using CSIS artifact is informationally equivalent to SIS design developed using security enriched use cases.

3.6. Evaluation

During evaluation stage, artifact support to solving the problem is observed and measured. In addition, the evaluation stage must determine whether the artifacts meets validity criteria. Evaluation involves comparing the objectives of a solution to actual observed results from use of the artifact in the demonstration (Peffers et al., 2007). Qualitative action research is used to ensure the artifacts rigor. At the end of evaluation stage, it must be decided whether to iterate back to design to try to improve the effectiveness of the artifact or to continue on to communication and leave further improvement to subsequent projects.

The initial relevance cycle defines acceptance criteria for the ultimate evaluation of the research results. The output must be returned to the environment for study and evaluation in the application domain. A design artifact must be evaluated to demonstrate its validity, utility, quality, and efficacy (Gregor & Hevner, 2013). Validity means that the artifact works and does what it is meant to do; and, is dependable in operational terms in achieving its goals. The utility criteria assess whether the achievement of goals has value outside the development environment (Gregor & Hevner, 2013). Because design is inherently an iterative and incremental activity, the evaluation phase provides essential feedback to the construction phase as to the quality of the design process and the design product under development (Hevner et al., 2004).

Hevner et al., (2004) suggested that the nature of the problem, characteristics of the artifact, and available resources should drive the selection of the evaluation method. The goal of evaluation is to establish that the design artifact fulfills the requirements and constraints of the problem domain and therefore it is complete and effective. A rigorous design evaluation may draw from many potential techniques, such as analytics, case studies, experiments, or simulations (Hevner et al., 2004). A summary of possible evaluation methods and their applicability to the CSIS design artifact is listed in Table 4.

An observational case study and field study are viable evaluation methods for the CSIS artifact. Observing the use of the artifact allows the researcher to validate if the artifact achieved its goal of securing the organizational business process in IS design successfully and in an efficient manner. Such validation can be conducted if the design artifact can be studied in a controlled experiment to study if the secure design is operational and usable. However, simulation with artificial data may not be sufficient for evaluating the artifact based on its socio-technical nature.

Analytical evaluation methods such as static analysis and dynamic analysis may not be suitable for evaluating the artifact. The design artifact is, in essence, a design method based on knowledge as operational principles (Gregor & Hevner, 2013). Static and dynamic qualities would not validate what the artifact is supposed to achieve. With sociotechnical artifacts in IS, the design is complex in terms of the size of the artifact and the number of components (social and technical).

Table 4. Design Evaluation Methods Applicability

Type	Technique	Description	Applicable to study?
Observational	Case Study	Study artifact in depth in business environment	√
	Field Study	Monitor use of artifact in multiple projects	√
Analytical	Static Analysis	Examine structure of artifact for static qualities (e.g., complexity)	X
	Architecture Analysis	Study fit of artifact into technical IS architecture	X
	Optimization	Demonstrate inherent optimal properties of artifact or provide optimality bounds on artifact behavior	X
	Dynamic Analysis	Study artifact in use for dynamic qualities (e.g., performance)	X
Experimental	Controlled Experiment	Study artifact in controlled environment for qualities (e.g., usability)	√
	Simulation	Execute artifact with artificial data	X
Testing	Functional (Black Box) Testing	Execute artifact interfaces to discover failures and identify defects	X
	Structural (White Box) Testing	Perform coverage testing of some metric (e.g., execution paths) in the artifact implementation	X
Descriptive	Informed Argument	Use information from the knowledge base (e.g., relevant research) to build a convincing argument for the artifacts utility	√
	Scenarios	Construct detailed scenarios around the artifact to demonstrate its utility	√

Architecture analysis has been developed for the kernel theory behind the artifact. However, architectural analysis may not be sufficient for the artifact based on its socio-technical nature. Black and white box testing may not be sufficient for the same reasons. Focus groups have gained increased attention in the IS field as a research method (Torkzadeh et al., 2006; Smith, 1996). Also, the software engineering discipline has suggested the use of focus groups as an evaluation technique (Kontio et al., 2004; Rouge & Niederman, 2006).

A focus group is defined as a moderated discussion among six to twelve people who discuss a topic under the direction of a moderator whose role is to promote interaction and keep the discussion on the topic of interest (Stewart & Shamdasani, 2014). The term focus refers to the fact that the interview is limited to a small number of issues. The questions in a focus group are open-ended but are carefully predetermined. The set of questions or questioning route is meant to feel spontaneous but is carefully planned. Usually, the moderator encourages the sharing of ideas and careful attention is paid to understanding the feelings, comments, and thought processes of the participants as they discuss issues (Krueger & Casey, 2014). A typical focus group lasts about two hours and covers a predetermined range of topics. Multiple focus groups allow for understanding the range of opinions of people across several groups and provide a much more natural environment than personal interviews because people are allowed to interact, which allows them to both influences and be influenced by others (Krueger & Casey, 2014).

A form of use case scenario has been applied during demonstration stage. Classical evaluation techniques do not seem suitable due to the socio-technical nature of our design artifact. A suitable way of evaluating the artifacts is "*go into the world and try them out*" (Baskerville, 1997). An action research (AR) environment has been made available to the researcher. Action research provides a way to improve the practical relevance of IS research and not just theoretical aspects. This research focused on the use of action research for artifact evaluation as explained in the next section.

3.6.1. Artifact Evaluation using Action Research

The contribution of the secure IS design artifact lies in its utility in the organization where its applied. In the CSIS artifact, utility lies in the artifact relevance to IS development. This research contended that current secure information system design methods suffer from applicability issues to IS development and software development methodologies. Action research aims to solve current practical problems while expanding scientific knowledge (Baskerville & Myers, 2004).

It has been described as "*the touchstone of most good organizational development practice*" and "*remains the primary methodology for the practice of organizational development*" (Baskerville & Myers, 2004). Therefore, action research may be helpful in determining whether the CSIS design artifact is applicable for good organizational development practice. It is in the context of the rigor versus relevance dilemma that action research holds particular appeal (Mårtensson & Lee, 2004).

Action research is a qualitative research method used in social sciences (Myers, 1997). Baskerville describes action research as an "*interventionist approach to the acquisition of scientific knowledge that has sound foundations in the post-positivist tradition*" (Baskerville & Wood-Harper, 1996). It was proposed for studying human groups and dynamics from the perspective of bringing about change in society. Unlike natural sciences e.g. physics and mathematics, social sciences suffer from replicability issues since social phenomena is not homogeneous through time (Peter Checkland & Holwell, 1998). Action research was introduced to address difficulties faced by social scientists who would like to make use of the outstanding successful method of inquiry developed in the natural sciences (Peter Checkland & Holwell, 1998).

Action research avoids the limitations of studying complex real social events in a laboratory by engaging the researcher himself or herself in a human situation and following it along whatever path it takes as it unfolds through time in a real environment. This means that the only object of research becomes the change process itself. There has been increased attention on the use of action research as research method in information system research (Peter Checkland & Holwell, 1998). The information system field is becoming more concerned with the social aspects of introduction of technology into the human workplace, rather than focusing solely on the technical aspects (Siponen, 2002).

Action research allows the researcher to intervene the practitioner environment and introduce change in systems development methodologies. This makes action research an ideal method to evaluate the utility of the CSIS design artifact. It is not possible to study a newly invented technique without intervening in some way to inject the new technique into the practitioner environment, "*go into the world and try them out*" (Baskerville, 1997). Also, the obtained knowledge about secure information system design can be directly applied in an organizational environment based on clear conceptual work linking theory, practice and research objectives (Susman, 1983). Evaluating IS research using action research is described in the following section.

Action research might intervene to introduce enhancements by means of change. Changes must have theoretical implications outside the organizational environment. Theory building takes an incremental and cyclic form. The kernel theory in the design artifact was utilized (Markus et al., 2002). The secure information system design method is based on organization based access control (OrBAC) (Kalam et al., 2003) as the kernel theory. This research drew modeling concepts from contexts described in OrBAC. Research in an organization on how to introduce a secure information system design evolved into research on what organizational changes are first needed to make it possible to introduce a secure information system design.

In order to achieve scientific rigor using action research, this research followed action research phases described in Baskerville (1999) which includes diagnosing, action planning, action taking, evaluating and specifying learning (see Figure 6) in an iterative manner.

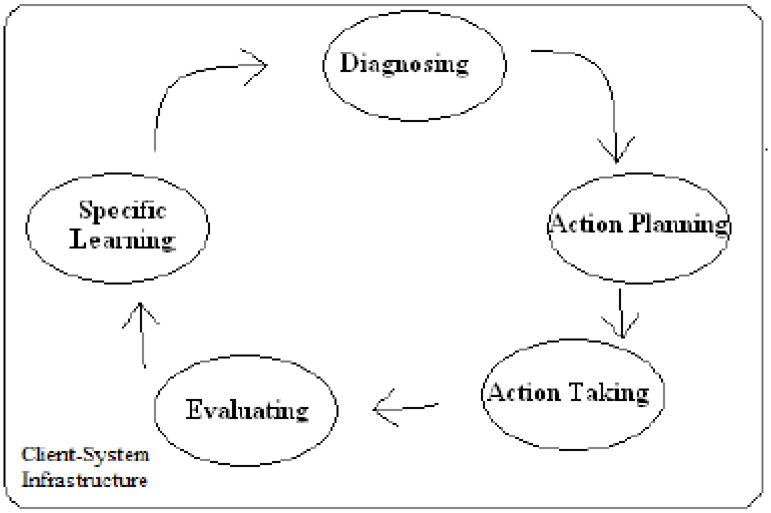


Figure 6. The Action Research Cycle (Baskerville, 1999)

The Client-System infrastructure is the specification of the research environment. It describes the client and the researcher responsibilities to each other. Working closely with IS practitioners involved in the study provides key insights on the phenomena under study (Baskerville, 1999).

The five action research phases constitute the research environment under which the researcher and practitioner take specific actions. (Baskerville, 1999) defines these phases as follows:

1. *Diagnosis* involves identifying the primary, complex organizational problems that organizations want to change in a holistic manner. In this phase, the researcher develops certain theoretical assumptions about the nature of the organization and the problem domain.
2. *Action Planning* goal is to identify organizational actions that would relieve or improve primary problems. The plan establishes the target for change and the approach to change. Action planning involves collaboration between researcher and practitioner to specify actions to improve or correct the problems identified in the diagnosing phase.
3. *Action Taking* implements the planned action. Action taking is a collaborative, dynamic phase. IS practitioners and researchers work together causing certain changes. The intervention might be directive, in which the research "directs" the change, or non-directive, in which the change is sought indirectly. Intervention tactics can also be adopted can include direct modification of process or procedures, indirect hiring of experts, or other intervention strategies.
4. *Evaluating* occurs once the planned actions are completed. Evaluation includes determining whether the theoretical effects of the action were realized and whether these effects relieved the problems. The evaluation must critically question whether the action undertaken where the change was successful was the sole cause of success. If the change was of no success, researcher cycles through

another iteration after adjusting hypotheses or design approach. The researcher and practitioner determine if the problems were corrected and if the theoretical effects of the actions were realized.

5. In the *specifying learning* stage, results of research are directed to three audiences. The organization will adjust norms to reflect the newly gained knowledge in SIS design. If planned action was unsuccessful, additional knowledge provides foundations for diagnosing in preparation for researchers to conduct further research interventions. Whether the theoretical framework was successful or not, important knowledge would be gained from the scientific community with future research settings. In the diagnosis phase, this research attempted to pinpoint core problems that cause the information system to be insecure and understand the reasoning behind the organization's desire for a change of its' IS design. The diagnosis took place in a generic holistic fashion.

The action planning phase is also a collaborative task in which this research specified the changes or actions that would improve the information system design. The plan described the targeted organization and the approach to be taken. A theoretical framework (F) guided this planned change, which is a collection of the organizational desired state and the set of changes that would achieve this desired state.

The third or action-taking phase is an implementation of the planned change. The researcher and the practitioners indulged into action in the client organization leading to a change in the organizational IS design. The researcher directed the change to ensure that it is taking its course within the organization.

Next, this research conducted an evaluation phase in which the researcher verified if the effects of the actions taken took effect and if those effects alleviated the security issues in IS design. In addition, the researcher made sure that the action taken was the sole cause of the success in the design. If the change was for some reason unsuccessful, a cyclic iteration of action research starts. The research started the first cycle when the intervention at the organization started.

The cyclic iteration stopped when the evaluation criteria were met as discussed in the following sections. The final learning phase provided the emergent knowledge to the IS research community. If the evaluation results were not successful, the learning phase will provide insights to conduct further action research interventions. Whether the result is successful or not, action research cycles can continue to develop further emergent knowledge about the organization and validate theoretical assumptions.

The action research process cycle is described in Figure 7. This process is based on Checkland (1991) model of “organized use of rational thought” which describe the action research cyclic process where F is an intellectual framework of linked ideas (a theory), and M is a methodology for using this framework. Moreover, A is an area of application. Action research cycles the research themes of F and M through A to generate reflection, action and finally scientific findings. The real world problem and research settings are described first; action research phases are then described in the following sections.

3.7. Research setting and organization selection

Action research was conducted in a Fortune 500 software business organization. This organization is an American multinational software company that provides different server, networking, and applications.

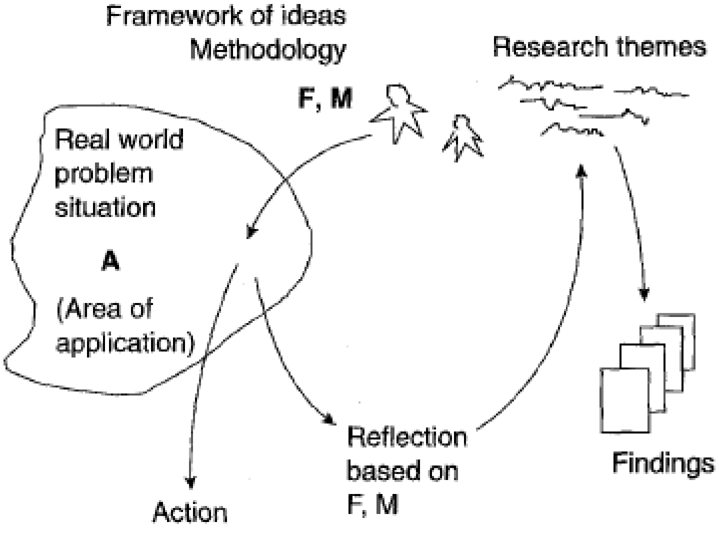


Figure 7. Cyclic Process of Action Research (Checkland, 1991)

The organization development centers reside in different geographical locations in Florida, California, Australia, Canada, Denmark, Germany, India, China and the United Kingdom. The organization ranks high on the Fortune 500. The organization reports more than $3.5 billion in revenue with hundreds of millions in net income every year. More than 10,000 employees work for this organization.

With this spread presence across the globe, the organization is vulnerable to various cyber threats. There is a need for an information system that meets the organization development requirements. The security of said information system must be ensured. The organization was selected since the researcher worked there. The researcher had access to organizations sensitive security information during the research. There were existing security issues in the organization in terms of code management. This helped set up a suitable environment for action research.

One major business process for a software corporation is to deliver software products where product source code is built, signed, packaged and delivered. This includes a wide infrastructure of servers, internal networks, source code management systems among other systems such as payroll systems, collaborations systems etc. The business process requires participation from different IS stakeholders. The spectrum of stakeholders includes developers, staff engineers, build engineers, infrastructure engineers, and project managers.

The software development process supports a continuous integration pipeline. The process includes specifying requirements for the software product followed by a coding stage. Then the code is stored in code management systems. The next stage is building the software then testing it. Finally, the product is packaged and delivered to customer systems.

A critical asset for a software organization is its source code. There are several different software development projects conducted within the organization with millions of lines of code. The organization faced problems when managing source code access for different software projects in a secure and efficient manner. There are more than 10,000 employees working on different projects with different source code access needs along with thousands of automated servers that require daily access to some parts of the source code. The existing code management system was unable to identify the right permissions to grant for a given user or system.

An example would be involving a software development project manager and an infrastructure engineer in granting specific access to some code for a developer. Details on the organizational problem are discussed in detail in the diagnosis stage in the next section. This research drew from relevant action research processes in IS and contrasted different stages of action research in Table 5.

Table 5. Overview of Action Research Processes

Action Research	Puhakainen & Siponen (2010)	Börjesson et al., (2006)	Lindgreb et al., (2004)	This Research
Client	● One organization	● One organization	● Six organizations	● One organization
Diagnosis	● 16 Interviews ● Open Ended Questions ● Anonymous Surveys ● Observations	● Send survey to all employees ● Observations	● Technology review ● Workshop sessions ● Discussion and analysis with practitioners and researchers	● 8 interviews ● 6 open-ended questions ● Observations ● Discussions and analysis with practitioners and researcher
Action Planning	●Designed program based on problems found in diagnosis stage	● Collaboration with reference group	● Develop design principles	● 8 interviews ● 7 open-ended questions ● Collaborated with work group on requirement analysis for a design of code management system based on organizational requirement following CSIS development.
Action Taking	● Training delivered ● CEO meeting ● Issues addressed	● 2 Meetings with strategically chosen group members ● Apply proposed process ● 4 meetings with Reference group as opinion leaders	● Implement design principles	● Develop design for code management system for distributed code repositories. ● Meet with strategically chosen group members who managed old systems.
Evaluation	● Users interviewed (personal and group) ● Process improvement observed	● Interview and feedback collection	● Focus group ● Participant observations ● 24 semi-structured interviews ● Open and axial coding technique	● 8 interviews ● 14 open-ended questions ● Feedback collection ● Process improvement observed ● Open coding technique ● Focus Group Evaluation ● New system users interviewed in Focus Group
Specifying Learning	● IS security policy compliance	● Agile improvement practices	● Competence model	● Secure information system design method ● Context risk analysis

3.7.1. Diagnosis

With the intent to identify a secure information system design, the research started the first action research cycle by investigating the participating organizations earlier attempt to secure its information systems and assets. Developers are the most common users of source code. However, infrastructure engineers manage source code repositories. Build engineers administered code repositories and granted permission to different repositories.

A collection of systems collaborated to enable the main process of this organization, which is producing software. The organization used a system for managing ownership and access to various source code repositories. Different actors in the organization log into this system using their organizational credentials to interact with source code systems. Managers, in turn, received requests to access source code repositories and create permissions according to what they see fit.

The existing problem was that the code management system was unable to handle permissions correctly based on the software development projects running. The system required manual intervention from an infrastructure engineer and a project manager to apply a change on source code access permissions where in effect only a project manager should be involved in applying change permissions for a given developer.
Users are required to ask administrators for specific permissions that might be then revisited by project managers. In additions, applying code permissions in certain countries requires certain care. There are specific rules for code access in specific counters. This usually includes access forms that need to be signed.

The code management system was impeding the software development process. The organization decided to switch to a distributed model for source code repositories. However, the security of distributed source repository is questionable within the organization.

There are concerns on how source code management might be applied and how to secure code permissions. In the diagnosis phase, the researcher worked with practitioners responsible for developing, managing and maintaining the existing source code management system. This group in the organization was chosen as they are the experts on the existing code management system. The researcher synthesized semi-structured interview sessions to capture existing issues within the current system. Also capturing concerns and perceived problems on the design of the existing systems.

The researcher asked introducing, probing and follow-up questions to better understand the phenomena and the problems in the existing systems (Kvale & Flinders, 1997). The researcher started with introductory qualitative research interview questions and then possible follow-up questions (Kvale & Flinders, 1997). The researcher drew these questions from relevant action research in IS and contextualized them to this research study. The questions are described in Table 6. The researcher conducted interviews with experts responsible for existing systems. The researcher then analyzed the answers and conducted discussions with practitioners.

The open-ended questions are in line with qualitative data analysis technique (Strauss & Corbin, 1998). IS researchers have approached data collection similarly. Questions asked in the diagnosis stage in Puhakainen & Siponen (2010) were contextual to the study where they were attempting to plan IS security training. They are aimed at understanding existing awareness problems such as “Do you think security instructions in the organization useful?" and "In your opinion, what are the most common ways malicious software (viruses etc.) gets into our company's network?".

Table 6. Diagnosis Action Questions

No.	Question	Supporting Literature
1	what are the most common ways malicious software (viruses etc.) gets into our company's network? What are the information security problems in the existing system?	Puhakainen & Siponen (2010)
2	Where can you find our company's official information security instructions? How are permissions granted to users?	Sedlack (2012)
3	Are there any other security issues that you consider important for your work? Do developers need managers and admins to be granted access?	Puhakainen & Siponen (2010)
4	How often is there personnel or manual intervention involved in granting code permissions ?	Follow up on diagnosis question 3
5	Do you think about information security only when interacting with information security programs or objects? What kind of access violations to source code may occur?	Sedlack (2012)
6	What are various groups that are granted access to source code?	Follow up on diagnosis question 5

In this study, similar open-ended questions that are relevant to the security of the design of the information system were asked in order to understand the existing problems. The suggested questionnaire was sufficient to address different aspects of existing IS design. Other questions did arise during following action cycle. Follow up and probing questions were risen as the semi-structured interviews took place.

3.7.2. Action Planning

Following diagnosis phase, this research followed CSIS principles in providing a secure approach to source code management system design that fits well with the organizational agile requirements. The researcher first collected the new code management system requirements from the expert group using CSIS principles. System requirements were then defined with CSIS in mind. Semi-structured interviews were conducted with the focus group. The interview questions are described in Table 7.

The researcher presented the design plans within the same group responsible for the management and maintenance of existing code management system. The goal in this stage was to develop a design for a secure code management system using CSIS. Therefore, the questions were directed towards collecting the context construct data for the design of the new proposed system. These questions were chosen for use in the planning the design of the information system.

Table 7. Action Planning Questions

No.	Question	Supporting Literature
1	What actors are involved in interacting with source code?	Kalam et al., (2003)
2	What context are these actors involved in order to have the need to interact with source code repositories?	Kalam et al., (2003)
3	What are the spatial contexts when interacting with source code?	Kalam et al., (2003)
4	What are the temporal contexts when interacting with source code, if any?	Kalam et al., (2003)
5	What are the prerequisites contexts when interacting with source code?	Kalam et al., (2003)
6	What are the provisional contexts for interacting with source code?	Kalam et al., (2003)
7	What are custom contexts when interacting with source code?	Kalam et al., (2003)

The basis of these questions is the kernel theory OrBAC in which the researcher attempted to derive the contexts for the proposed IS design.

3.7.3. Action Taking

Following the action-taking plan using CSIS design principles, the researcher worked with practitioners to implement a prototype for a secure code management system leveraging distributed code repositories. The first action was to form a group to develop a prototype of the code management system. Based on the contexts collected in the action-planning phase, the researcher defined the way the new code management systems works following requirements analyzed in the previous meetings.

The researcher conducted meetings with strategically chosen group members who managed old systems in order to evaluate the utility of the new system. This group acted as an opinion leader in the implementation of the code management system. They oversaw the development and suggested changes as they saw fit. The goal in this stage was to produce a prototype design for a secure code management system.

3.7.4. Evaluating the Results

Upon the completion of a prototype, the focus group and users of new code management system design prototype were interviewed. The introduced change was assessed and evaluated by collecting feedback on the utility of the new system. The interviewees were asked open-ended design evaluation questions described in Table 8. These questions were selected to validate whether the designed secure information system meets Siponen's et al., (2006) meta-design requirements and whether the theory behind the design was validated through successful use (Baskerville & Myers 2004).

Action research must ensure that the intervention was the sole cause for the success of the design (Baskerville, 1999). The researcher observed for any process improvement in security and permissions management in the new system. Questionnaire results were classified, interview results and meeting data were then compiled Börjesson et al., (2006), Puhakainen & Siponen (2010). Positive feedback on the use of the new system was then observed by soliciting feedback from participants. Other cycles of action were initiated based on the feedback received on the developed design prototype and the qualitative data analysis.

3.8. Data Analysis

Data was collected by means of open-ended questionnaires, interviews, observations and field notes. The interviews, field notes, and observations were transcribed on the same day into an electronic word processing file. The names of individuals were anonymized using codes for purposes of transcripts and data analysis to protect the identity of the participants.

Grounded theory techniques to qualitative data analysis were followed. Data analysis phases using coding techniques are outlined here:

- *Open Coding*: The transcripts were analyzed and codes were assigned to the data via a qualitative analysis software. Each interview transcript was examined and codes were applied to the appropriate selection of text as interpreted by the researcher. Once the interpretive, descriptive and pattern codes were applied, any emergent themes that arose were analyzed.

Table 8. Evaluating Action Questions

No.	Question	Supporting Literature
1	Does the design resist the impact of threats on system objects?	Siponen et al., (2006)
2	Does the design meet organizations' security requirements?	Siponen et al., (2006)
3	Does the design provide abstract representation and operations for specifying the three essential elements of secure systems - threats, objects, and security features (safeguards or controls) - for the three levels of abstraction: organizational, conceptual and technical.	Siponen et al., (2006)
4	Can the design be easily integrated into normal information system development methods?	Siponen et al., (2006)
5	Does the design maximize the autonomy of developers?	Siponen et al., (2006)
6	Is the design adaptable to forthcoming information system development methods?	Siponen et al., (2006)
7	Does the system enable DevOps agility?	Follow up on evaluation question 6
8	How do you feel that information security improved in the project?	Sedlack (2012)
9	What changes enhanced information security in the project?	Sedlack (2012)
10	Do you feel project members better understand information security relating to the project? Why?	Sedlack (2012)
11	Do you feel this project remains aligned with organizational goals while providing balanced information security? Why?	Sedlack (2012)
12	Did the research model help improve member information security perspectives relating to the organization? How?	Sedlack (2012)
13	Did the research model improve project information security? How?	Sedlack (2012)
14	Would you change the model for the next project iteration, adding or removing elements? Why? How?	(Sedlack, 2012)

- *Axial Coding:* Following intensive open coding, identified categories were brought together into groupings. Axial coding was used to determine the relationships and patterns between the various codes. Additional comments and code maps were developed to assist the researcher in the visualization of patterns and themes contained within the interpretation of the data. These

> groupings resemble themes and are generally new ways of seeing and understanding the phenomenon under study.

The theoretical framework must guide the analysis (Rubin & Rubin, 2011). The codes were driven by CSIS design constructs. Patterns and themes arose from the analysis to provide comparison and contrast to determine conceptual explanations of the study. The source material was grouped together using the identified codes. The resulting design was examined to determine if the application of the theoretical framework addressed the research problem.

3.9. Validity Criteria for Action Research

Action research must have a practical outcome (Baskerville & Myers 2004). This corresponds to the internal validity of action research (Baskerville & Wood-Harper, 1998). In addition, action research must have an explicit underlying theory before conducting research; it must demonstrate a clear contribution to research (Baskerville & Myers 2004). This corresponds to the external validity of action research.

To ensure the validity of this research, Baskerville & Wood-Harper (1998) validity criteria for IS action research were applied:

- The research should be set in multivariate social situations.
- The observations should be recorded and analyzed in an interpretive frame.
- Researcher actions should intervene in the research setting.
- The method of data collection should include participatory observation.

- Changes in the social setting should be studied.
- The immediate problem in the social setting must have been resolved during the research: The current organizational problem must be resolved during the study according to the evaluations made by focus group.
- The research should illuminate a theoretical framework that explains how the actions led to a favorable outcome.

3.10. Summary

Leveraging design science paradigm, CSIS artifact was presented. The novelty of the artifact is the use of context in IS design. It allowed bridging a research gap by explaining how a socio-technical secure design approach can be integrated into actual IS design. A demonstrative example captured supporting evidence for the first research hypothesis. This research introduced a method and investigated how to integrate socio-technical aspects in IS development stages using the context in IS and organization. The process of evaluation using action research was described along with data collection and analysis processes. The following chapter describes the cycles of the organizational action research.

Chapter 4

Journey to Cloud Cadence – Action Research

4.1. Introduction

The empirical part of this dissertation concentrated on exploring the truth and utility of the CSIS design artifact presented in chapter 3. The empirical study was conducted within a software organization. Because security is a sensitive subject for the company, all identities have been disguised.

The rest of the chapter presents the organization throughout the transition towards a new Source Code System (SCM) and a new SCM management system. The design of a new SCM management system as per the CSIS design artifact is described in section 4.2. The next section describes the action research cycles (Baskerville, 1999). This includes interactions between the researcher and practitioner throughout the process. Finally, focus group discussion analysis is described.

4.2. Adopting a New Source Control Paradigm

The researcher started seeking approval for the study with the organization in May 2016 and was engaged from June 2016 until December 2016 at SC located in the southeast region of the United States. SC is one of the Fortune 500 software business organization. This organization is an American multinational software company that provides different server, networking, and applications. The organization development centers reside in different geographical locations in Florida, California, Australia, Ukraine, Canada, Denmark, Germany, India, China and the United Kingdom. More than 10,000 employees work for this organization.

SC organization was attempting to secure its assets including source code and application data. During the same time, SC was transitioning from Agile (Börjesson et al., 2006) to DevOps (Loukides, 2012) engineering to adapt to a cloud-based release cadence. The number of software releases in the cloud is much higher than regular on-premise product releases. There is a higher number of changes to adapt to user requirements in the cloud. SC organization must increase its release cycles to adapt to a continuously changing market. It has become almost an industry standard to adopt a DevOps approach. One of the main challenges is the changing nature of requirements for the design. The project manager described the problem

> *"it's hard to design a system for today when you know it's going to change. I know the security requirements are constantly changing."*

Microsoft went on a similar journey towards a cloud-based distributed version control system. SC attempted to capture lessons learned from Microsoft. Microsoft described its journey to transition its processes and shared their experience online (Guckenheimer, 2016). SC organization planned migration from centralized version control systems to distributed version control systems as part of the cloud cadence transition. In the following, a brief description of version control systems is described and how the migration is perceived in the organization is presented.

Version control systems are divided into two groups, centralized and distributed. Centralized version control systems such as ClearCase, CVS, Subversion (SVN) or

Perforce are based on the idea that there is a single centralized copy of the project source code. Usually, the code is hosted on a code server, and the IS users submit their "changes" to the server.

Distributed version control systems like Mercurial and Git do not necessarily rely on a central server to store all the versions of a project’s files. Instead, every users "clones" a copy of a repository and has the full history of the project on their own hard drive. This copy (or “clone”) has all of the metadata of the original. The IS users can "commit" their changes to their local copy until they are comfortable to "push" their code changes to other users.

There are several advantages with using distributed version control systems over centralized systems. Performing actions other than pushing and pulling "changes" is faster because the user only needs to access the drive, not a remote server. “Committing” new code changes can be done locally without anyone else seeing them. Once a group of code changes is ready, all of them can be “pushed” at once. Everything except code “pushing” and code “pulling” can be done without a connection to the organization, which gives more flexibility to system users. IS users can work remotely, temporary code changes need not to be submitted. Changes can be “stashed” if the user wants to save their work at a certain state.

However, permission control can be tricky since users have their own copy of the whole code. Security concerns stem from the openness of distributed version control systems.

One disadvantage of distributed control systems is that every user will have a full copy of all code history. This issue raised security and usability concerns.

All IS users would have a full copy of the code including its history which raised security concerns to IS administrators. In addition, the existing system stored binary files within the project source code in large amounts. The distributed system approach was not suitable for the existing system since all copies of binary files would be stored on the user's hard disk making it full.

In this context, distributed systems have an advantage over centralized systems in that they offer the ability for the user to create their own (lightweight) version of the code locally to develop a certain feature. For example, a topic branch might be created per work item and cleaned up when the changes are merged into the mainline. This allows developers autonomy when developing product features (Siponen, 2002).

However, distributed version control systems were largely questioned from a security perspective in SC organization. Because of their distributed nature, they allow developers to clone a copy of the whole code branch he\she is viewing. Permissions in a centralized version control system can be as granular as a single file as opposed to a whole branch in a distributed version control system. The participants perceived this as an information security gap in code management in what was considered more granularly controllable in terms of view and permission control in a centralized version control system as opposed to what the transition would introduce.

When SC relied on centralized version control systems for its software development purposes, a security audit in the organization highly recommended a well-defined business process. A participant commented,

> *"So the way it works now is we have a tool. So we went for a security audit a little while back – a couple of years ago – and they recommended, you know, having sort of a double approval for creating accounts and granting access to codes. So what we ended up doing was we wrote a tool that would allow you to raise a request for an account and then have your manager approve that request and then have an administrator approve the request and then the account is created. It's a really, really good tool. Very awesome 'Sarcasm'."*

SC corporation was dependent on a management system and some semi-automated process that was developed within the organization to manage its many instances of centralized version control systems. The researcher intervention started with introducing a design for a management system that helps adapt to the transition to a distributed system following CSIS.

4.3. Background and Participants

During the initial stage of action research, understanding the practitioner's perspective is essential. This requires analysis of a holistic organizational perspective. This section describes the analysis of organization interview questions.

The interview questions about a new system design were analyzed with CSIS design artifact in mind. Open coding was done using Atlas.ti v. 8.0.25 software. The researcher developed the design model for the management system by categorizing data through the artifact's constructs: Roles, Activities, Views and Contexts.

During action research cycles, Fifty-two meetings were held. These meetings included Twenty-Four one-on-one interviews and unofficial follow-up meetings. The meetings took place over 7 months, between the researcher and organizational personnel to understand existing systems security issues, identify the goal of the project, discuss new system design requirements, and provide insights during review stages of action cycles.

The interviewees were selected from a pool of people assigned to work on the implementation of the project and the transition phases. Out of sixteen people who worked on the project, eight participated in the research (see Table 9). Interviews and follow-ups took place in the workspace environment, and they were conducted during different action research cycles. The size of the company made it possible to interview all employees several times during the research process. The researcher then moderated a focus group discussion to confirm the utility of CSIS in comparison with SARC (D'Aubeterre et al., 2008).

4.4. Research Strategy and Position of the Researcher

In this action research, the researcher was not regarded as an objective, passive outsider. The researcher was a part of SC transition process. The researcher was involved in some of other technical aspects suggested in the project such as minimizing the size of code repository and enabling workflows to function within different subsystems. His further responsibilities included the planning and implementation of the new management system design during the second research cycle.

Table 9. Research Participants

No.	Participant Code Name	Organizational Role	Participated in study
1	G	Senior Product Development Manager	Yes
2	A	Life Cycle Management	Yes
3	D	Software Engineer \ Existing system Administrator	Yes
4	H	Software Engineer	Yes
5	E	Senior Software Engineer	Yes
6	F	Software Engineer	Yes
7	B	Senior Software Engineer	Yes
8	C	Senior Software Engineer	Yes
10	J	Principal Software Engineer	No
11	K	Principal Architect	No
12	L	Product Development Director	No
13	M	Product Manager	No
14	N	Senior Product Development Manager	No
15	O	Principal Software Engineer	No
16	P	Senior Software Engineer	No

The researchers' involvement for some of the design tasks was individual, but cooperation between the research and the collaborators was conducted in the study in describing and analyzing system design requirements (Baskerville & Wood-Harper, 1996). The action research intervention was aimed at providing a secure system design for distributed source control that meets the organizational security requirements and ensures the state of security in the organization. Another aim was to increase the security awareness of the participants through the use of the CSIS artifact. The business process objective is for the organization is to produce software products.

Information was collected and analyzed constantly throughout the research process using interviews. The interviews were conducted using open questions governing information security issues in the existing management system, and description of workflows. The questions were open, but detailed, aiming to explore employees' description of issues at state.

Three types of interviews were used. The participants were interviewed in normal social interactions and using formal interview techniques. The data was collected by means of interviews. Field notes were collected. In addition, some of the interviews were conducted online where the interviewer had an online video conversation while presenting questions and ideas to interviewees. This was done due to the international nature of SC. Some of the employees involved in the project were located in different areas of the world. Moreover, some of the interviews were conducted in informal social situations, the researcher had to follow up on some feedback that was given during a formal session.

When doubts arose about an interviewee's statements, it was verified immediately during the interview or later the same day to ensure consistent feedback to the action cycles. To avoid having an influence on the interview questions, the researcher started with general questions (Stringer, 2013) (e.g., "what do you think are security issues in the organization ?") and then followed up on details that focus on the system design (e.g., how do you think the process looks like in the new system?). The aim was to identify the themes that emerged from the data collected. Moreover, whether these themes supported the kernel design theory and the CSIS artifact. The analysis formed the basis for developing the intervention. One of the main challenges faced during design stages was the changing nature of security requirements of the organizational IS.

4.5. Source Control Management System Project

The transition from a centralized source control system to a distributed source control system project was selected by the researcher for action research. This project was selected for research because of its security requirements. The goal was to develop a management IS that support the security requirements of SC for distributed version control systems. This allowed the researcher to demonstrate the adaptability of the CSIS context to highly changing requirements.

In June 2016, the researcher met with project management to discuss possible candidates for the study. This action research consisted of five research cycles, which are described in the following subsections.

4.5.1 Research Cycle 1

There are five stages in each action research cycles. These are diagnosis, action planning, action taking, evaluations and specifying learning. In this cycle, initial interviews discussed existing security issues. Planning and action taking focused on the new IS design.

STAGE 1: IS Problems Diagnosis

In order to provide a secure system design for SC distributed version control systems. The system requirements must be captured. The first stage in the diagnosis phase is to explore the existing systems issues and capabilities and capture the participant's perspective on their systems security. The researcher aimed to identify current security issues, concerns, practices to help identify existing problems in the existing management system. The researcher asked questions described in Table 6.

The main theme of security concerns was related to employees remotely logging in with their personal devices and downloading source code. There was a concern of external devices logging in to the network as these devices cannot be controlled by the organization and they simply may be stolen. Also, injecting malware into the source code systems via USB sticks, downloading malicious software via web pages. In addition, there were concerns regarding phishing emails.

There seemed to be a lack of security awareness amongst participants of the study. Four out of eight participants had no idea where to find security-related information to the organization e.g. security design guidelines, security policies. Participants' comments included,

> "*I believe our security team may run our website. But if so, there's no requirements to go to it and I don't know that anyone ever does go to it*."

> "*It's probably scattered around. I don't know,*"

> "*I don't see any of that. I don't know where I could find it.*"

Specific to the existing management system, participants have monitored for possible malware being checked in into the system. A participant said,

> "*You could have people inject things that you don't want into your code, whether it's malware or viruses, Trojans, just things that lower the performance of the code even.*"

Another problem was the fact that user permissions get "stale" on the system. When a user starts using the system and then leaves the team, his permissions are not revoked. Stale permissions allow possible reentry to the system that might cause security incidents like theft or retaliation. Participants noted,

> "*Gaining access is request-based. And no one requests their access be removed. We accumulate access but never deprecate access, even if we happen to come across something that looks stale.*"
>
> "*Sticking to the existence of one of the weaknesses of the existing setup is that we accumulate but never remove permissions. I just today found permissions like from 2006 for a user that no longer exist in the organization.*"

The diagnosed issues indicate poor management in system user accounts. The management system was added as another layer of security by security auditors, it was not considered during initial stages of system design. There are different manual processes that are poorly defined when it comes to managing users. A participant said,

> "*every time someone needs more access or a new account. Then at some point a admin or a build team members acting in that role is going to be involved. I would say that's at an absolute minimum several times a week and more likely several times per day that that's happening. It's a little hard to get a quantification of it because there are probably eight or nine different people who may respond to them and take care of a given request. So those requests are being spread out across the team, essentially. And I don't monitor what each person is doing for each little thing – like that little*"

In addition, four out of eight participants complained about the lack of auditing on such permission changes. The next stage of the action cycle is to fully define the different organizational roles in order to alleviate the user management problem. The researcher identified roles in the organization based on the participant’s descriptions. The identified roles that were found are, Developers, Architects, Testers, Development Leads, Gate Keepers, Escalation, Graphics, Publications, and Engineering Services. In the next subsection action, planning for the system design is discussed.

STAGE 2: Action Planning

The second phase of the first action research cycle is planning a new management system design. The research framework for system design followed CSIS artifact. Artifact constructs were described to the participants in order to collect data in the form of system requirements. Essentially, the participants were asked questions in Table 7. The questions focused on defining Roles, Activities, Actions, Views, Objects, Contexts from the participant’s perspective.

Answers varied on contexts, participants provided different contexts for the business processes they described. This helped enrich the design artifact with various contexts. Themes emerged from the participant’s definitions of the design constructs. These themes are described here.

Subjects and Roles at SC focused on the software development process given the organization nature. However, the organization works in a collaborative manner in order to deliver the software to the customer. There are software architects, software testers, graphical designers and technical writers. There are also software components owners, development managers, gatekeepers, and leaders along with software continuous integration build team and a source control system management team. Finally, there are technical support and escalation engineers.

Notably, these roles are replicated across all geographical locations at SC. However, there are different regulations in each country. This makes the design requirements different for different roles in different countries. The existing system replicated these roles for geographical locations in order to enforce such rules, which created another source of confusion for user management. There is also non-engineering roles in SC such as legal, accounting and HR. However, since they do not interact with the system they were not considered in the design.

Activities and Actions at SC also focused on the development process. Developers at SC work on developing software product features and fixing bugs in existing software releases. Architects look for design patterns in the software. Testers ensure the quality of the software and report bugs; they also work on automating test cases for the software. Graphical designers work on creating the graphics for the software UI and technical writers work on writing software documentation. Software component owners and managers ensure that software features they own are delivered on time and they also work on managing development cycles. Gatekeepers ensure that software changes included in the next release meet some sanity and testing criteria.

The continuous integration build team is responsible for wrapping the code into a software package that can be delivered to the customer. They also handle the life cycle and maintenance of the product. The SCM team ensures the existing centralized source control systems are always running and making sure they are always backed up. Technical support and escalations report customer feedback and analyze their input based on their knowledge of user manual and documentation.

All of the activities described so far are considered read and write actions to system objects. Another action called “approve” was considered as a special case of “write” where a code moderator approves code changes. System admins, component owners, and build engineers are permitted to “delegate” permissions to other users in the systems. System admins are responsible for all “creation” actions including, the creation of users in the existing management system, “creating” new code repositories in existing version control systems. In addition, all system roles are permitted to “log in” to the system to initialize their view.

Views and Objects at SC augmented various assets including source code and infrastructure. The majority of the work at SC is being done on source code files. The files are stored in code repositories, which are hosted on source code systems. The identified system objects are the hosting servers and the code repositories. The views on hosting servers are defined as software view to the virtual machines running on servers and physical view to physical access servers. The code repositories are organized in terms of project names and in some cases in terms of the objective to the role. For example, there is a code repository for graphics and a code repository for technical documentation. However, each software component is code repository.

Repositories and folder paths can be branched (i.e. copied with a new name with a pointer to the “HEAD” repository), these repositories can be used to release software or lock at a certain point for a certain released version and supporting that repository for maintenance. For example, software release “X” branch, legacy release “Y” or current product branch “Main”. Similar concepts are used to derive the system views. A view is constructed for projects developers are working on, a view for graphics branches, a view for documentation branches and also views for main and released code branches.

System and Business Process Contexts at SC were identified during planning. The participants were introduced to the context concept in the action planning stage. They defined different context to be used in the design of the system following different contexts available in CSIS artifact.

As a prerequisite context, developers are only allowed to submit code if and only if the code compiles. Another prerequisite context was that the code must pass secure function checks e.g. use secure printf functions. Also, manager approval is a prerequisite for developer’s ability to access project code.

In terms of spatial contexts depending on the subject location, a restriction was enforced on certain geographical locations. Developers in certain geographical locations had to sign a non-disclosure agreement to ensure the confidentiality of software source code. In addition, spatially, users remoting into the management system must be connecting through virtual private network to the corporate network. In addition, user locale must be ensured when connecting to the system ensuring that users are connecting to the servers residing in their own country.

In terms of user-declared contexts, depending on the subject objective the participants discussed many contexts. First, two-factor authentication when logging in to the management system. Second, ensuring that account is created when logging in to the management system. A target is assigned to work on a project. A user joins a team or a user's switches teams or leaves the organization. Finally, previous releases are accessible to the user when attempting to view new releases. In the following action-taking phase, the design of a management system for distributed version control for SC is described.

STAGE 3: Action Taking - New System Design

Using the emerged themes from analysis of coded data. During the third stage of the first action research cycle, the researcher constructed the organizational IS design using CSIS artifact. Then the researcher presented the artifacts to the participants for review and evaluations. The design artifact is described in Appendix C.

STAGE 4: Evaluation of Design

The fourth stage of the first research cycle was to evaluate the resulting system design artifact. The participants were introduced to the design artifact of the management system. The interviews focused on explaining the design artifact to participants and then collecting feedback. The interview questions also discussed meeting (Siponen et al., 2006) secure IS meta-design requirements described in Table 8. From IS practitioners point of view that participated in the study, the design meets meta-design requirements for secure system design.

The participants agreed that the design artifact meets the organization security requirements (Siponen et al., 2006). A participant stated,

> "*it seemed to have all the requirements we had built in*"

On an organizational, conceptual and technical level, safeguards and controls are described as participants have noted,

> "*I think objects and security features are certainly visible in the diagram across organizational as well as conceptual and technical. I think you covered that for objects and security features. I see the countermeasures to the threats in the diagram. For example, hacking in from outside the company. That's blocked by you have to be in a VPN or in one of our offices.*"

> "*We have enough information to safeguard whatever is required*"

The participants agreed that the design artifact described the requirements for developing the secure system. The design can be integrated into existing development methods; it also enabled DevOps agility. It can be adaptable to IS development (Siponen et al., 2006).

> "*I mean, it looks like there's a lot of requirements that are shown in this diagram – use cases and things that need to be developed within in. So like that would work pretty well.*"

> "*It has separated branches which can be developed in parallel. So in terms of the development method I'm thinking it should not be very hard to fit it into the current development method*".

The design also enabled developer's autonomy (Siponen et al., 2006). A participant commented,

> "*I certainly think it would. I think there's separation there. And then you could easily figure out a list of what needs to be done to support the system from the diagram.*". Another participant also said, "*I think as much as you could say is that it gives them autonomy so long as they stay within the framework*".

The participants reasoned that the design is adaptable to forthcoming information system development methods (Siponen et al., 2006).

> "*It's fairly generic, nothing tied to a specific technology*"

> "*Yeah, I think it's fairly generic. There's nothing tied.*"

> "*Yeah, I mean, I don't think this design matters, right? That's the beauty of the methodologies is that they can be applied to any type of design. Really the methodology is about the team that you have and what people are comfortable with and how they do development stuff.*"

The participants felt like information security improved in the project (Siponen et al., 2006).

> "*Design of a proper workflow. Capturing and documenting the Intended use of the system is a huge improvement. Also, enforcing context that were never enforced before.*"

> "*you've considered a lot more around the overall security of the system.*"

> "*Yes, there will certainly be consistency and process that there hasn't been in the past and that's a good thing. You know, because this should in principle have caught everything that we currently deal with and we will actually now have, you know... That consistency is important in security, but obviously it's implementing that and actually ensuring that it's adhered to.*"

> "*I think having things that are a bit more automatic and things that are a bit more, you know, context aware... Certainly things like, you know, I have to be a member of this group and I have to be at a certain location, I have to be in certain business hours and stuff like that, like that's more powerful than we have today so I think that's definitely much better.*"

Along with information security improvement, the balance between security and the organizational objective was maintained. Participants commented on the design,

> "*I thought it was pretty balanced the way you just asked that question. I think it definitely looked at what do people need to be able to do and then how do we make sure that we are protecting against threats in doing that. And I would say balance is the best thing you can have in terms of security, personally.*"

> "*... absolutely this process should definitely aim... head towards that.*"

The design process also improved information security perspectives relating to the organization as perceived by participants.

> "*Yeah, you sure have some advantages to improve with because we gathered together to brainstorm about a lot of possibilities.*"

> "*Yeah, so that gave me like a lot of different insights on like at the time of designing a piece, right. So more of the various aspects just to be kept in mind. Like the different scenarios, the different actors and their roles, how*

> *they're interlinked to each other, and one role can like be superseded by the other and so on. So those are the various information which you and I did gain from the discussions what we had.*"

> "*The design made me think about security. The design model used roles, subjects and objects, I suspected these terms come from security, I had to look it up.*"

STAGE 5: Specifying Learning

Capturing the security requirements along with functional requirements during design helped achieve improvement in the project. In addition, a core aspect was capturing business process context for security improvement and awareness. A participant said*,*

> "*The context in which people should be allowed to perform activities. I do think contextually aware security is definitely the future of security and needs to be included in the model like you've added here*"

The extensibility of contexts was attractive to developers, a participant said,

> "*I think over time obviously you'd probably want more in different contexts and stuff like that. So I think there would be additions going forward. You might want a little more logging or something. I mean, these are things that we can add to the system as time goes by*"

Despite positive results, the need for further development remains. There were issues identified in the design artifact. The designed entailed complexity, it was prevalent more for specific roles. Participants noted complexity in the design,

> "*I'm thinking this system it looks like complex; however,*".

> "*yeah, even we have different separated branches can be developed in parallel, like chop them to different pieces. However, we still have complex role with a lot of…*"

There were issues providing helpful implementation hints to developers of the system. A participant said,

> "*I think there is an extra step in between of taking this diagram and breaking that into… Obviously you can get requirements in there very easily from that diagram, but there's still the breaking that into action items that I think is a separate thing.*"

The initial cycle resulted in an initial IS design. Based on collected feedback, the design needed revisiting to alleviate complexity.

4.5.2 Research Cycle 2

It was evident that the initial design resulting from the first action cycle was complex. The second cycle focused on revising the design to simplify readability. During this cycle, a revised design was presented to participants.

STAGE 1: New System Design Diagnosis

A secure design for a management system of distributed version control systems was presented to participants in the first action cycle. The researcher asked questions about the utility of the design. The collected feedback indicated high complexity in the design. The participants reported readability issues for complex roles.

> "*Like this section – the build section I think you have on the screen now – is an incredibly tangled mess down there right now*"

> "*We still have complex role with a lot of…*"

> "*If there was a way to highlight a role when clicking on it...*"

The researcher agreed that complex roles have complex artifacts that are hard to read. The overall artifact provides a holistic picture of the system design, but it makes it hard to investigate a specific business process. This, in turn, was impeding readability of the artifact.

STAGE 2: Action Planning

The holistic artifact was big in size. In order to increase the readability of the artifact and alleviate the complexity of the system design. The researcher decided to chop the artifact of the design based on the role following participant's feedback, also maintaining the full artifact on the overall system for reference. There should be no changes in design semantics but rather present the work in a more focused manner.

STAGE 3: Action Taking

The researcher worked on refining the new system design into components based on role. Even though the full artifact provides a holistic view of the system, the researcher added artifacts depicting workflows for all roles to ease the presentation of the design. The researcher took into consideration the feedback resulting from the first cycle. The design artifacts are presented in Appendix C.

STAGE 4: Evaluation

The researcher kept the full design artifact intact and added separate artifacts that are more narrowly focused on the business process for all organizational roles. The artifacts helped shape a user story that can be developed. This allowed for more agility for possible system implementation. Participants felt like DevOps approaches would be applicable to implement the design. A participant said,

> "*I think we can use Kanban to track that*"

STAGE 5: Specifying Learning

There still was arguing opinions about the effectiveness and information richness of the new design artifacts or the split artifacts. A participant said,

> "*I don't know if I like the fact that you split everything apart. I mean, it's easier to focus in on certain areas but it's harder to see the whole. But I think that's just personal preference.*"

> "*I was going to say I actually quite like the fact it's broken up because then you don't get too distracted by, you know, other arrows and other, you know, things going on…. Yeah, other things going on*"

The initial artifact was more suitable for complete system analysis and the added artifacts were used for creating user stories. Despite the improved presentation of the system design. There still was concerns about the semantic understanding of the workflow. A participant noted

> "*… still all I see is circles and arrows. I know you did explain what they mean those but if you did not…*"

4.5.3 Research Cycle 3

The revised artifact from the second research action cycle alleviated complexity. However, the design still needed revising. In this research cycle. The design is adjusted to follow literature notation (D'Aubeterre et al., 2008). This helped differentiated design constructs from a participant perspective.

STAGE 1: Diagnosis of SC Management System Design Artifacts

All constructs were depicted in the artifact and the design model using the same shape. The items in the resulting design look the same. This caused misinterpretation of the design artifact. The constructs of the CSIS design artifact must be clearly defined in order to convey the semantics of the design.

The researcher decided to revise the design artifact. The participants agreed that the artifact represents the requirements of the business process but it is still rather confusing, as all components look the same. This research adopted D'Aubeterre et al., (2008) notation in the design artifact. The revised design artifact enforces labeling constructs in the resulting design artifact. In addition, actors, roles, and objects use a special shape to distinguish them among other constructs. However, since there is no defined shape for context in IS literature, this researcher adopted pentagon to distinguish it amongst the others.

STAGE 2: Planning Revised Revisions of CSIS Artifact and Design Model

The researcher worked on revising the CSIS artifact to adopt to the changes identified in the diagnosis stage of the action cycle. The revised CSIS artifact is introduced in Figure 8.

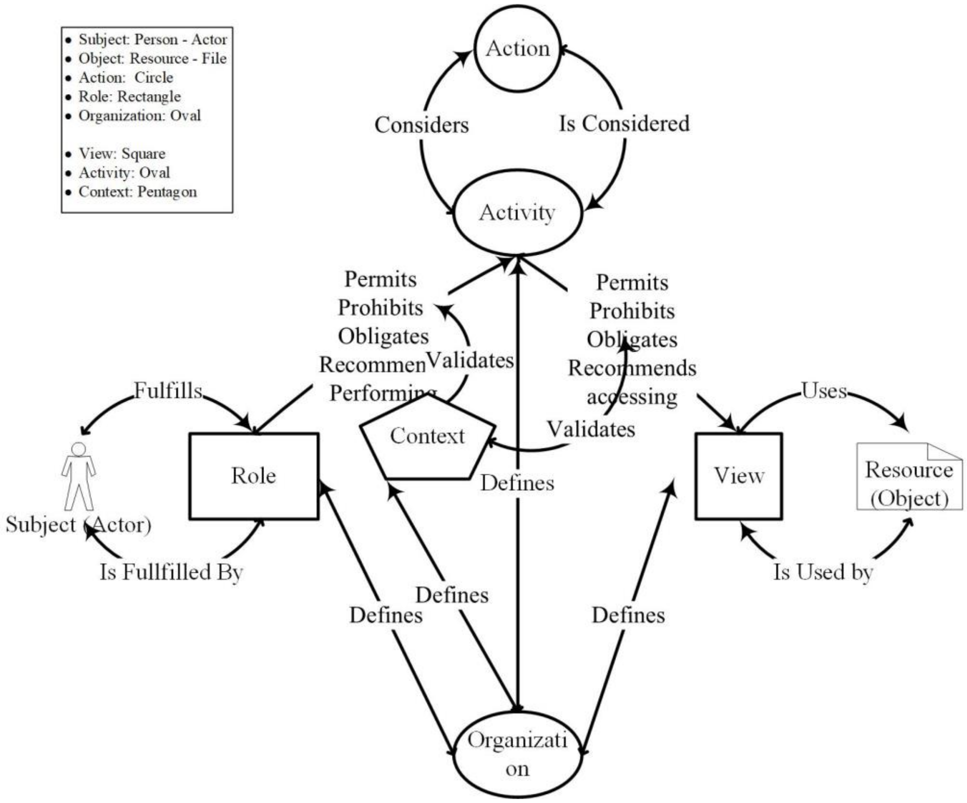


Figure 8. Revised CSIS Artifact

STAGE 3: Action Taking

The researcher refined the organizational IS design based on the revised CSIS artifact. There was no further feedback from participants at this cycle except the addition of labeling. The design artifacts are presented in Appendix C.

STAGE 4: Evaluating the Revised Design at SC

The revised design was clear to participants. They were able to easily identify and distinguish between Context, Actors, Roles, and Views. The resulting artifacts helped alleviate confusion and increase readability of the design. From a research perspective, the design artifact aligns with IS literature. From a practitioner's perspective, the revised design is much easier to understand.

STAGE 5: Specifying Learning

The participants felt more confident about the success of implementation after they felt more in control of the system design. The artifacts looked much clearer and easier to capture and understand. After analyzing the business processes workflows in the resulting design artifact and during a brainstorming session, it seemed obvious to the researcher and participants that there are particular contexts that need more attention from a security perspective. For example, the developer submitting code that does not compile is not necessarily a high-security threat. However, logging in the system without having a user created should be considered a high-risk threat. These ideas provoked the researcher to revise the design artifact to highlight these ideas.

At the end of the third research cycle, the design met IS and sociotechnical requirements. However, further information seemed to have revealed in the revised artifacts. The finding were then analyzed in the next cycle.

4.5.4. Research Cycle 4

After multiple iterations of the new system design. The participants developed a focus on the security aspects of the system design. The introduction of context allowed participants to think of possible system violation scenarios. Those violations were diagnosed in this cycle.

STAGE 1: Diagnosis of Risks in System Design

During the third action research cycle, it was noticed that there are different levels of risks associated with different contexts in the resulting design artifact. This was iterated back to the CSIS artifact. The CSIS artifact was extended to include risk levels to possible attack vectors that are associated with business process contexts.

Since possible attack may occur by violating system contexts. System contexts monitor over possible attack surfaces of the system. However, the violation may not be of high risk. There are low, medium and high levels of risks that may be associated with a context in the workflow description.

STAGE 2: Identifying Risks in CSIS

To identify the risk level of context violation, the CSIS artifact must identify the level of risks for context violation. To identify and highlight high-risk levels in the CSIS artifacts colors are used. The design artifact for SC was revised to include risk level for business process contexts. Red was used in high-risk contexts, Yellow was used for medium level risk contexts, Green was used for low-level risk contexts. A participant noted,

> "*Yeah, in the background, though. But what I can do is… I had the same thought. But I was thinking maybe light them up in colors*."

STAGE 3: Revising the design at SC

The researcher revised the design artifacts at SC that resulted from the third action cycle. The researcher identified the risks and highlighted with the designated colors. The contexts associated with activities related to security operations were perceived to have the highest risks. For example, log in operations, permissions changes would have higher risks than user submitting invalid code. The resulting designs are presented in Appendix C.

STAGE 4: Evaluating the Results

The participants felt that the introduction of colors makes the system developer careful when implementing workflows. Risks are higher than the cost of implementing these features if the context is violated when the system is implemented. No further modifications were suggested.

STAGE 5: Specifying Learning

Context as any other artifact construct can be assigned a risk weight. A secure system design can include risk factors during design stages. This is in addition to security requirements during design stages. No further changes were suggested to the CSIS artifact or the resulting design model.

4.5.5. Research Cycle 5

The final artifact was presented to participants. All were satisfied with the resulting artifact. Since the artifact and the intervention solved the organizational problem, this ended action research in the study. There is room for improvement for CSIS in future research endeavors.

During initial action research cycle, the researcher diagnosed existing IS security issues. Then planned, designed and reviewed initial design for new IS using CSIS. Feedback was then collected.

In the second action research cycle, the artifact was perceived as monolithic and complex. Issues were diagnosed and the design was adjusted to divide the artifact into multiple components highlighting the workflow for organizational roles. Feedback was then collected.

In the third action research cycle, design constructs were still being misinterpreted as there was distinctions in the constructs depicted in the artifact. The issue was diagnosed and the design was adjusted to follow D'Aubeterre et al. (2008) notation. Pentagon is suggested for context as there is no standard in literature. Feedback was then collected.

In the fourth action research cycle, certain contexts violations were perceived by participants to have a higher risk than others were. The observation was diagnosed and the design was adjusted to highlight the risk level of context violation on the design using color (low, medium, high) (green, yellow, red) respectively. Feedback was then collected. During the fifth action research cycle, no further adjustments were suggested, the final design was accepted to the organization and the action research cycles iterations stopped. Focus group was then initiated.

The utility of the CSIS artifact was evident during action research cycles. Collected feedback helped verify that the resulting artifact is information equivalent to that of SARC (D'Aubeterre et al., 2008). This, in turn, showed evidence to support the second research hypothesis in the study. Following, the researcher conducted a focus group amongst all IS practitioners participating in the study to discuss and evaluate the design artifact.

4.6. Focus Group for Design Artifact Refinement and Evaluation

There were no further iterations as no changes to the CSIS artifact or the design were suggested. As the design solved the organizational problem, the researcher needed to further validate the research hypotheses. To ensure the rigor of this research, the focus group was used as a confirmatory method to test hypotheses (Tremblay et al., 2010).

The SARC design artifact was introduced to the participants during a focus group discussion. Focus groups allow for an open format and are flexible enough to handle a wide range of design topics and domains (Tremblay et al., 2010). This helped create a rich amount of qualitative data about SARC and CSIS from an IS practitioner's perspective.

Planning a Focus Group

To compare if SARC and CSIS are informationally equivalent, The researcher used the same data that was collected and coded for constructing the CSIS design artifacts to create equivalent SARC artifact. The researcher presented these artifacts to the participants after explaining the SARC artifact during the focus group. The resulting SARC artifact is presented in Appendix C.

The researcher conducted a focus group to include all participants that were interviewed individually in previous action cycles. The meeting was conducted online to accommodate the international nature of SC. Some participants had to connect during their nighttime to join the discussion. The focus group meeting was conducted during business work hours locally. A pretest questioning route (Tremblay et al., 2010) during focus group meeting discussed on meeting secure system meta-design criteria (Siponen et al., 2006). Comparison between CISD and SARC in terms of representing the system requirements was conducted.

As a final step in planning the focus group, Tremblay et al., (2010) suggested finishing the focus group with a task where the focus group participants are asked to utilize and evaluate the artifact. Possible users of the system of the system participated in eliciting the requirements for the system, they were asked to evaluate the system design. The researcher worked on implementing the system workflows related to developers and administrators using the artifacts that resulted in the fourth action researcher cycle.

The researcher implemented the prototype as an independent web service that monitors over distributed version control systems operations and validates business process contexts. The system also implements the workflows as described in the design artifacts along with defined contexts. The researcher presented the new system early prototype to the participants during the focus group.

Conducting the Focus Group

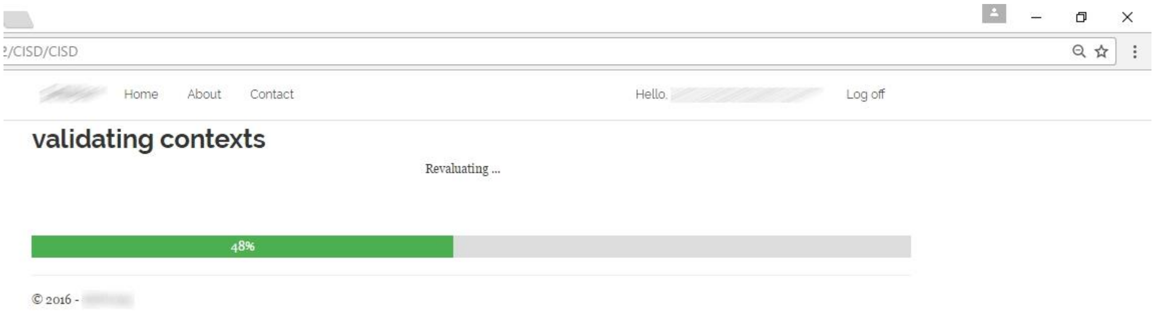
Based on observation, participants joined the meeting with a positive attitude. The team was eager for the transition towards a new system, which created excitement (Tremblay et al., 2010).

All participants were presented with SARC and CSIS artifacts beginning with an explanation of the motivation and theory between both approaches. Participants were then presented with different organizational scenarios and designs using CSIS and SARC.

Following an explanation of different scenarios on where and how the CSIS and SARC could be utilized, a description of the details of SC organization IS design using CSIS and SARC was discussed. Design artifacts for SARC and CSIS were presented.

Finally, users were presented with the early prototype to examine and utilize. The system works independently of distributed version control system. However, logging it to the system allows for re-evaluating system contexts. This can be noticed in Figure 9a, it shows an IS administrator logging in to the system.

The context status for the admin is valid. However, overall IS context is invalid, a user is attempting access to source code prior to signing NDA agreement as shown in figure 9b.



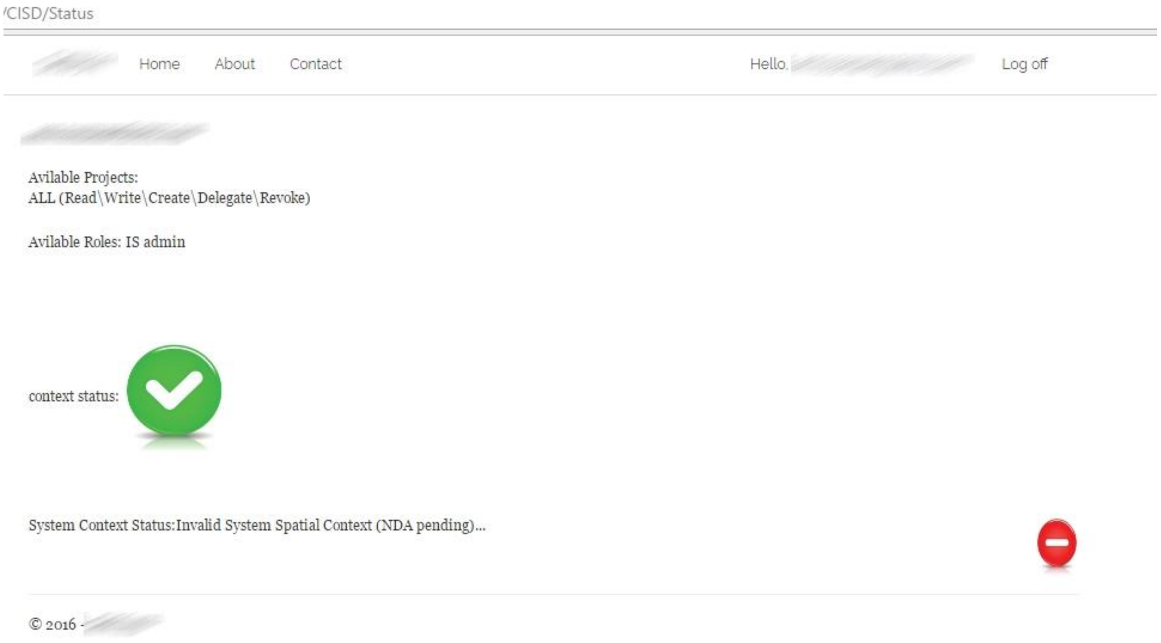
validating contexts

Revaluating ...

48%

© 2016 - [redacted]

Figure 9a. Management System Prototype for Distributed Version Control System Based on CSIS Design Artifact



[redacted]

Avilable Projects:
ALL (Read\Write\Create\Delegate\Revoke)

Avilable Roles: IS admin

context status: ✓

System Context Status:Invalid System Spatial Context (NDA pending)...

© 2016 - [redacted]

Figure 9b. Management System Prototype for Distributed Version Control System Based on CSIS Design Artifact

The administrators' contexts are valid; however, the overall systems contexts are invalid since there was a user that needed to fulfill a spatial context requirement. An email alert is already sent to IS administrators reporting the incident.

Evaluating the Results

In terms of security control, the participants felt that there is no description of security controls while describing secure business processes in the SARC artifact.

> "*It just feels like a very simplistic view to me, where it's not really even including security. It's just saying developers use the code repo.*"

> "*Now, isn't this security like…? So it's role-based security, right? So you log in as an administrator you get access to certain things. You log in as a user you get access to a different set of things. You log in as a backup user you might have a different set of things. You log in as a reviewer, different… So this is all pre… So this is similar to CSIS's, but I think you can think of it like the contexts are all predefined and they cannot change*"

Applicability to IS development was discussed during the focus group for both CSIS and SARC. Participants felt like both designs are implementable however, they did not feel that SARC offers much in terms of security

> "*I would say this would actually be implementable, but it would be a very insecure system.*"

Since objects are used from a low-level perspective, this restricts granularity in defining permissions on objects, unlike CSIS artifacts which can build up a view of projects.

> "*Well, I see like there's sub design that maybe isn't reflected here. Like, for example, a developer gets access to code. But if he's an X developer he might have different code versus an Y developer versus someone who's an architect; they might want to look at lots of code.*"

Utilization of the implementation showed improvement in permission management for the new distributed version control system in the organization. There is no longer a "special" unknown process in place, it all can be seen in one place.

> “*What I do think is better about this compared to even old implementation is everything is in one place now versus, for example, with the China developers they have to sign a form. All of that was managed completely external to the old centralized version control system itself. It involved emailing the admins and getting a form signed in some third system and then admin says it's okay and finally they're allowed to go and have a system account. It was a lot of overhead outside of the system that's not even… The system had no idea any of that's happening. I think now it is better that your implementation as well as your design kind of brings it all into a single place, which just was not there right. It involved admins and I knowing there's some special process in place”*

The CSIS design artifacts were demonstrated to be equivalent to SARC. CSIS artifacts also added more security controls over the system design. CSIS artifacts helped achieve a secure transition to a new management system to distributed version control. The designs helped increase security awareness amongst participants

> *“The design made me think about security. The design model used roles, subjects and objects, I suspected these terms come from security, I had to look it up.”*
>
> *“roles do not go straight into any activities they want but there's all those little green and yellow context boxes that are in play as well”*

The SIS design developed using CSIS artifact generated higher security awareness than SIS developed using SARC for experienced IS practitioners and security experts, which supports the third research hypothesis.

One of the meta-design requirements for secure system design (Siponen et al., 2006) “SIS design methods should provide modeling support at the three levels of ISD (organizational, conceptual, and technical) for abstract representation and operations to specify the three essential elements of secure systems: threats, objects, and security features (safeguards or controls).”. The resulting design at SC may appear to weakly support this requirement since it did not directly describe threats to the system e.g. phishing emails or possible malware downloaded to the system. One of the participants commented,

> “*I didn’t see anything in the diagram related to the threats that exist. But I don’t see the actual threat of an outside attacker shown in the diagram.*”

However, the design described security features in terms of safeguards and controls on all levels. The design introduced risk measures to areas where threats may occur. A simple attack tree (Cipriano et al., 2011) can be extracted from the design artifact. The project design required technically skilled practitioners with abstract design thinking. However, one of the participants was not technically skilled and so did not see the design of either SARC or CSIS as applicable to IS development. Referring to SARC complexity during design, a participant said,

> “*I cannot believe someone published this*”

They thought that these artifacts do a good job at defining requirements and they are necessary but not sufficient,

> “*if we were to just take this and send it off to somebody to implement, its may not be sufficient for implementation…. implement it, And secondly, I don’t think the diagram has anything to do with indicating methods for implementation. So in implementation you could use, you know, a SQL database. You could use something that’s part of the tool if it was available. You could use a flat file. You could put all this data on some TechSpot. Like there’s nothing about any structures or storage or APIs or access for anything that is implementation oriented specified here. This looks to me*

> *like a diagramming of what the various types of access paths are, and this one is basically role-based. It doesn't even say, I guess… So yes, this shows here's the roles, here's what they can do. Could you implement this? I think this is necessary but not sufficient for an implementation.*"

That participant felt that they could not go from these artifacts straight to coding; they suggested possible use of classical flow chart artifacts and descriptions of "data structures" to supplement the design to provide a complete toolset for implementations. It can be considered if further UML can enable and supplement CSIS artifacts for implementation as long as they meet the security requirements.

Other participants felt that the artifacts are sufficient for coding. The inexperienced participant was a bit outdated on development methods, the project manager commented on the feedback provided by that participant,

> "*And I think a lot of us, or at least I personally was trapped in a lot of well, in the old thing, this happens, so that's a concern – even though we've probably fixed by now as well. I mean, I don't think that's any problem with the model or the design process you went through. I think that's a problem with the people participating and the project itself.*"

4.6. Discussion

The action research conducted at SC consisted of five research cycles. In the first research cycle, the intervention was the introduction of a system design for a management system. Research cycles were exploratory iterations that refined the design artifact and the system design. A focus group was conducted to confirm the utility of the design artifact and to test research hypothesis. Next, the results of these interventions are summarized and evaluated.

During this action research study, the researcher explored the relevance, applicability, and feasibility of the CSIS design to the organization. In addition, the researcher took an active role in solving a practical problem experienced by the host organization. The researcher worked on implementing a secure system that prevented IS security violations found during security audits and increased security awareness.

As part of its transition towards a cloud release cadence, the organization was moving towards distributed version control systems. The researcher worked closely with IS employees including participants and developers involved in managing these systems and subsystems to enable a transition to secure system design and processes. The CSIS design provided a guideline for implementing a management system for these systems. The researcher was responsible for developing secure IS design for the transition towards distributed version control during the first research cycle. His further responsibilities included acting as a system implementer.

Supporting evidence for the second research hypothesis was found in the intervention results. The focus group discussion showed that CSIS is more superior to SARC in utility and ensuring IS security. Supporting evidence for the third research hypothesis was found during the focus group.

4.7. Conclusion

This research was conducted to solve the information security gap in the existing research by developing a theoretically grounded and comprehensive approach on how to design a secure IS that can be integrated into all IS development stages. A software development organization was engaged in action research. Interviews were conducted with participants involved in the organizational project of moving towards cloud release cadence.

Action research was conducted to ensure the utility of the artifact in solving the stated problem and to test the hypotheses. The themes that emerged from the interviews were sufficient to conduct the design based on CSIS. The resulting design increased security awareness of participants as was evident during action research results evaluation.

According to interviews data analysis, the CSIS artifact meets Siponen et al., (2006) meta-design requirements. Supporting evidence for each of the meta-requirements was found and described during action research. Ensuring the security of the context in which people should be allowed to perform activities was found to ensure system security. Interviews also demonstrated that organizational context of business processes can be assigned a violation level. This helps identify high-risk attack surfaces in system design. The interviews helped include this design concept in the CSIS artifact.

Chapter 5

Conclusion

5.1. Introduction

This dissertation aimed at developing a theoretically grounded secure system design method. The CSIS artifact was developed to meet secure system meta-design requirements (Siponen et al., 2006) in a way that can be integrated into all IS development stages from requirements to deployment. An artifact was developed using design science paradigm. The truth and utility of CSIS artifact were evaluated via demonstration, action research and a focus group. This is summarized in Table 10.

Table 10. Design Science Research Validity Criteria (Hevner et al., 2004)

Design Science Research Validity Criteria	**Research Study**
Truth (justified theory or true statements corresponding with real world phenomena)	OrBAC as kernel theory (Kalam et. al, 2003)
Utility (building artifacts that are effective)	Action research Focus group Adaptability to IS design (Siponen et. al, 2006)

A new design model and method are presented for consideration. Then, a review of accepted evaluation criteria for qualitative research and how this study meets the principles is presented. The chapter concludes with a discussion.

5.2. Research Findings

In the first step, current SIS design approaches were reviewed. Analysis showed that current design models suffered from applicability issues to IS. Using design science research (Hevner et al., 2004), a design model was developed from a socio-technical perspective to enable applicability to SIS design. IS literature suggested that the most common organizational role of IS security, was socio-technical (Hirschheim & Klein, 1992), Baskerville (1992), Dhillon & Backhouse (2001), Siponen (2002) and D'Aubeterre et al., (2008). IS security is dependent not only on technical solutions but also on users' behavior.

As with literature, the CSIS artifact suggests that the organizational role in information security is socio-technical. The context of activities in which the CSIS design draws IS security from a socio-technical perspective. The truth and utility of the CSIS artifact (Hevner et al., 2004) were evaluated using two methods (See Table 10).

The first method is descriptive (using scenarios and informed argument). Consequently, the applicability of the artifact was tested using descriptive methods and was compared to Siponen's enriched use cases for equivalence. The CSIS artifact demonstrated its usefulness in capturing security requirements during early stages of IS design (see Section 3.5).

In the second method, observational evaluation was conducted where the artifact was utilized in an action research study. The artifact was used in an organization to prove its applicability to IS design. After several research iterations of applying the artifact in an organizational design, the artifact was revised until no more possible improvements were found. CSIS theory was adjusted according to practical outcome (Baskerville & Myers 2004). The artifact is described in Figure 10 below.

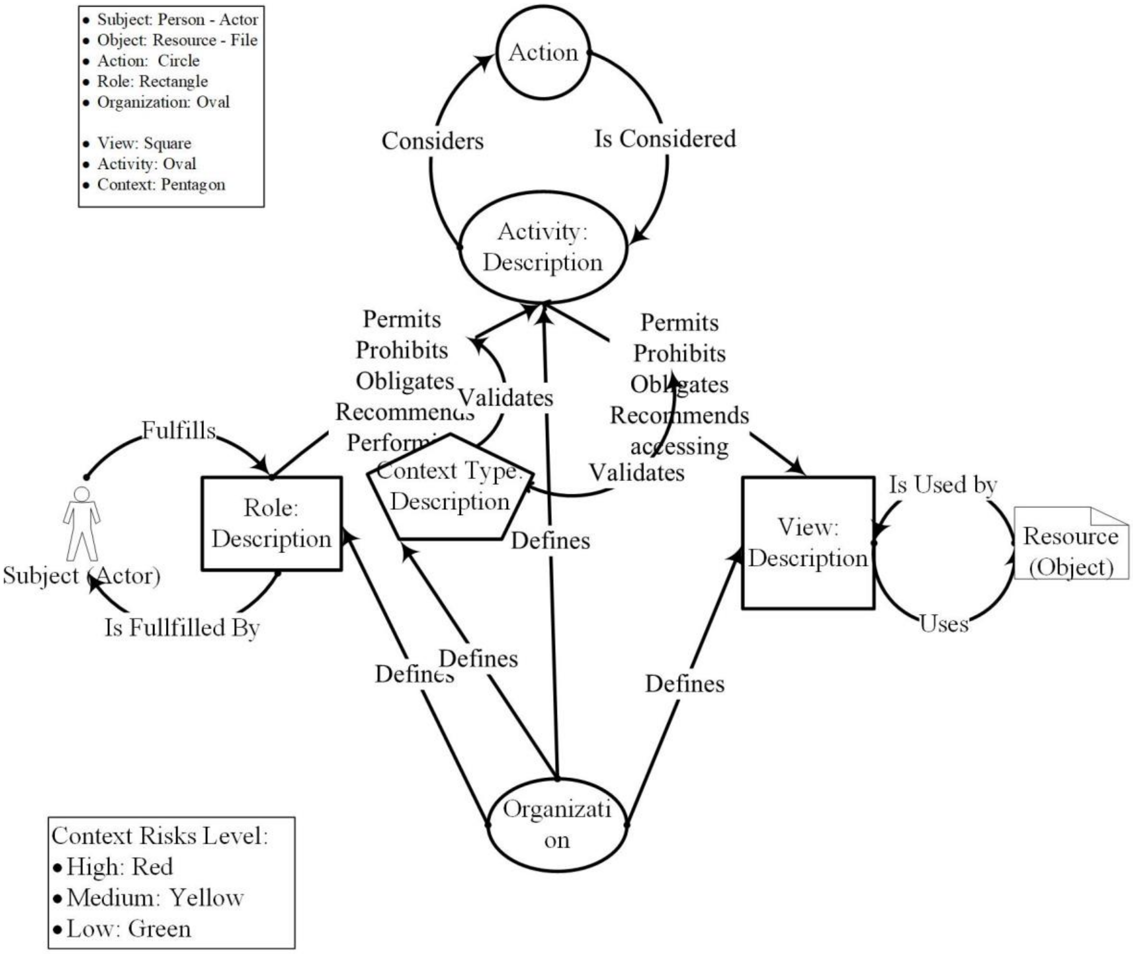


Figure 10. Final CSIS Design artifact

CSIS was found to meet SIS meta-design requirements (Siponen et al., 2006) when deployed in an action research intervention.

5.3. Research Results

Analysis of data collected from action research interviews and focus group confirmed the hypotheses are supported. The design was found superior in terms of securing IS assets in addition to information equivalence (Table 11). The CSIS artifacts provided different possible contexts for the business process activities, employees no longer needed to bend rules to achieve business objectives.

In addition, the CSIS artifacts provided sufficient abstraction to secure the system objects. Views provided greater separation amongst objects based on the objective. This information cannot be defined within a SARC artifact.

Table. 11 Research Hypotheses

	Hypothesis	Supported
H1	SIS design developed using CSIS artifact is informationally equivalent to SIS design developed using security enriched use cases or SARC.	Yes
H2	SIS design developed using CSIS artifact will generate higher security awareness than SIS developed using security enriched use cases or SARC.	Yes
H3	SIS design developed using CSIS artifact will generate higher security awareness than SIS developed using security enriched use cases or SARC for experienced IS practitioners and security experts.	Yes

Defining contexts when conducting system design activities increased the participant's awareness to the system security workflows. Using CSIS, IS security design takes a balanced view of end-user related and technical issues allowing for IS security awareness to have a balance as well. Future research should investigate other possible forms of context or constructs that heavily affect business processes contexts.

Assigning business processes context different risk levels helped highlight possible attack surfaces to the implemented system. The colors in the design indicated the level of threat if the context was violated. This was found during action research cycles and not during the original design of CSIS. Future research should investigate possible prevention systems based on context violations.

Action research was perceived to be a suitable research strategy for testing the CSIS design artifact and adapting the resultant design. To summarize, the action research experiences indicate that practitioners perceived the design model for SIS design as relevant and feasible. In addition, it was easily applied to development and deployment of a management system. Developing a secure code management system for newly adopted distributed code repositories enabled validating the utility of CSIS in an organizational setting.

5.4. Research Validity Criteria

This research study applied design science paradigm in addition to conducting action research. Therefore, this research must meet design science and action research validity criteria. Both aspects are discussed here.

Design Science artifact must prove to be useful in solving a problem (Peffers et al., 2007). The CSIS artifact proved useful in securing the information system. It can thus be seen as valid in terms of artifact truth and utility. The use of OrBAC as kernel theory provides evidence for the truth of the artifact. The demonstration provided in the third chapter describing booking scenario provides evidence for the utility of the artifact.

Action research interventions were successful providing utility in the organization and IS. We applied Baskerville &Wood-Harper (1998) validity criteria for IS action research to ensure the validity of research as described in section 3.6. See Table 12.

Table. 12 Action Research Validity Criteria Baskerville & Wood-Harper (1998)

Action Research Validity Criteria	**Research Study**
Multivariate social situation	Yes
Interpretive frame	Yes
Researcher actions intervened in the research setting	Yes
The method of data collection included participatory observations	Yes
Changes in the social setting were studied	Yes
The immediate problem in the social setting was resolved during the research	Yes
Actions are based on theory	Yes

- **The research should be set in multivariate social situations.** The action research was conducted with various employees of the company, involving various relationships between the participants.

- **The observations should be recorded and analyzed in an interpretive frame**. Existing IS users were interviewed during the problem analysis phase, during action planning and when the results of the action research cycles were evaluated. The interviews were stored throughout in the form of field notes. The field notes were then analyzed in an interpretive frame.
- **Researcher actions should intervene in the research setting.** The researcher worked actively and directly with the employees of the host organization. The researcher was responsible for designing and delivering the secure IS design. This entails that this research also complied with the requirement of practical action (Baskerville and Myers 2004).
- **The method of data collection should include participatory observation.** Interviews, participatory observations, follow-ups, and questionnaires were used. The researcher had the opportunity to spend some time at the company and gain approval to conduct research. This provided a good opportunity for participatory observation.
- **Changes in the social setting should be studied**. The outcome of the research was critically assessed with reference to the focus group views of the success of both the IS design success and security of said IS. Several employees were asked for the effectiveness of security on the newly implemented system. This means that this dissertation also fulfilled the requirement of socially situated action (Baskerville and Myers 2004).
- **The immediate problem in the social setting must have been resolved during the research.** A prototype implementation for a management system was

provided. The current organizational problem was resolved during the study according to the evaluations made by focus group.

- **The research should illuminate a theoretical framework that explains how the actions led to a favorable outcome.** The actions within the action research cycles are linked to OrBAC as kernel theory of our proposed secure information system design. CSIS provided the theoretical framework for designing a secure information system. A secure management system was a favorable outcome for the organization. This also means that this dissertation complies with the requirement of an explicit underlying theory before an action (Baskerville and Myers 2004).

The research findings support the research hypotheses in this study. This indicates that the design principle behind CSIS is valid. Contextually driven system design ensures system security. Context awareness in the design of the information system provides controls that help ensure IS security implementation. Implications of research for academia and practices is presented. Finally, limitations of research and proposed future research are discussed.

5.5. Research Contributions

The main contribution is a secure information system design principle that meets well-defined criteria (Siponen et al., 2006). This research introduced the use of context from a socio-technical perspective to IS design. In order to secure the processes of information system in an organization, it is fundamental to understand the context of organizational processes and activities. Secure system design must shift towards contextualism (Tejay, 2008).

This research introduced the use of CSIS artifact to develop secure systems. CSIS enables capturing sociotechnical and security aspects from as early as requirements stages of system design. System design using CSIS ensures system security.

This research extended the uses of an access control model to secure system methodology. This research used OrBAC principles of access control and applied them to develop secure systems. CSIS used OrBAC constructs to describe how contextual, emergent and changing requirements can be incorporated into the design of IS. CSIS introduced the use of context in OrBAC in IS design. It allowed bridging a research gap by explaining how a socio-technical secure design approach can be integrated into actual IS design.

This research introduced a better secure IS design artifact than SARC. CSIS is informationally equivalent to SARC. Moreover, SIS design developed using CSIS artifact will generate higher security awareness than SIS developed using SARC.

CSIS is an empirically tested artifact. Artifact use was demonstrated via use case scenarios. Action research and focus group were used to evaluate the design artifact. This helped ensure the artifact utility and the research rigor.

From a research methodology perspective, this research introduced the use of action research in design science. Multiple methods were used to satisfy design science validity criteria of truth and utility. This research adds to design artifact evaluation methods described in Table 4 (Hevner et al., 2004). The use of action research in evaluating the artifact allowed for testing the utility of the CSIS artifact. Use of focus group and interviews for evaluation phase of design science.

5.6. Implications for Practice

The CSIS artifact provides an easy design method for systems to IS practitioners. The CSIS design method can be easily adapted to IS development methods as the data analysis indicates from action research. The CSIS design artifact can also help the organization transition their development methods e.g. DevOps or any future development method as the CSIS design easily adapts to forthcoming methods.

The CSIS artifact is useful to organizations transitioning towards cloud cadence. System design using CSIS proved helpful in having control and visibility over permission management in distributed source control systems. System design with process context in mind will help ensure security in multi-tenant architectures.

5.7. Limitations

Design science produces new creative and innovative artifacts designed to solve an organizational problem (Simon, 1996). CSIS is a new creative and innovative that was empirically tested in an organizational setting for solving a problem. Since the research is of qualitative nature, the results are based on the researcher's interpretation. Research bias is part of qualitative research.

However, the aim was to minimize bias by assessing the utility of the system in action research and assessing the design in a focus group. Bias in this research was mitigated. The analysis focused on the research problem and validating the argument. The research was thoroughly validated from both action research and design science perspectives.

Action research does not aim to find general or universal mechanistic-causal laws. Rather, it aims at the provision means to take systematic action to resolve specific problems in practice (Stringer, 2013). However, at the same time, the relevant design methods need to be validated through successful use as was the case in this study.

In the action research processes described in this research, the goal was to solve practical problems experienced by the host organizations and to understand them and the results achieved from the viewpoint of theory. While action research system implementation results may not be generalizable, but they are of use in the host organization. In addition, the CSIS design artifact is utilizable in similar organizations as a point of departure in designing a secure system.

5.8. Future Research

While the context construct was well utilized in the CSIS artifact. The researcher still believes there are still further research investigations to be conducted with business process contexts. The CSIS artifact can be further validated in a different organizational context. Future research should investigate other possible forms of context or possibly constructs that heavily affect business processes contexts. Risk levels of system contexts can be great indicators of system attack. Future research should investigate possible prevention systems based on context violations.

5.9. Conclusion

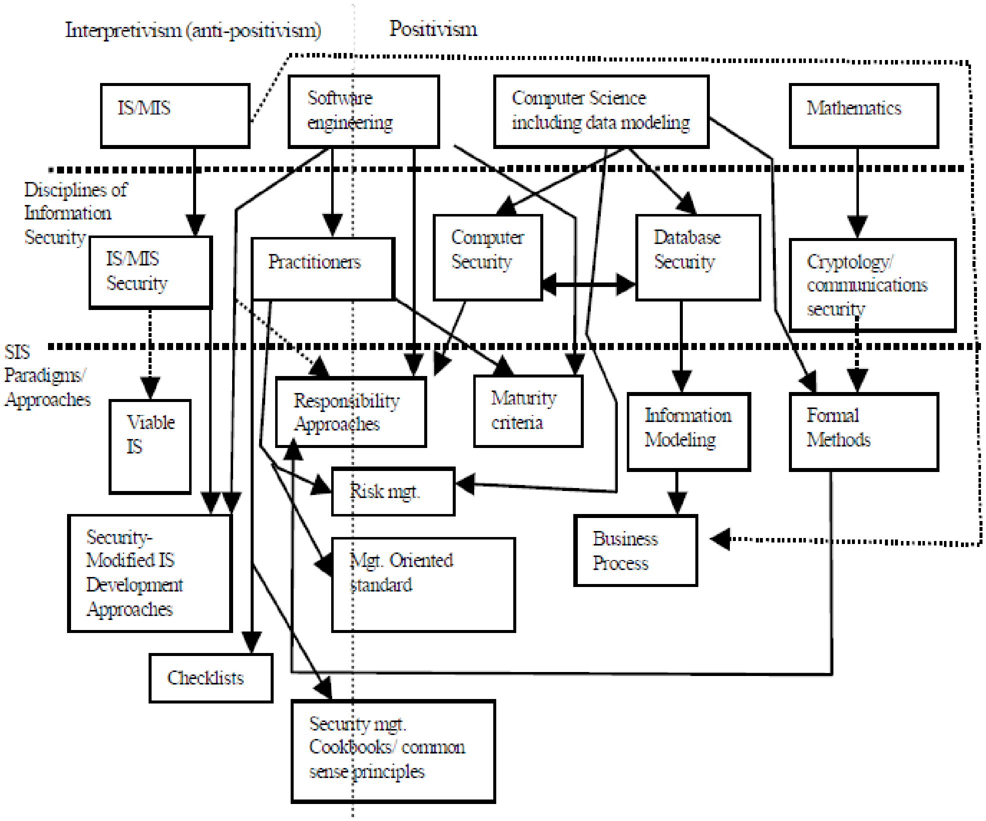
This dissertation consisted of two research steps. In the first step, since existing design models that captures socio-technical aspects of the system design suffer from applicability issues to IS, the researcher investigated how to design a secure system that considers security requirements from early stages of IS design. Following the design science research paradigm, this research built and evaluated an artifact designed to meet this driving business need and provided applicable knowledge to secure business processes. The introduced artifact captured the security role in the organization. The role of security in the organization is socio-technical.

In the second step, action research was conducted in an organization to ensure the utility of the artifact. The design artifact was revised during five research cycles. Research findings indicated the possibility of weighting attack risks based on a business process context. Using the artifact helped improve the security awareness of the users. A focus group was conducted to validate that the artifact does not miss information that may be included other approaches in IS literature to secure business process design. The interventions also indicate that CSIS meets secure system meta-design requirements.

For scholars, the use of context in IS design was introduced. It allowed bridging a research gap by explaining how a socio-technical secure design approach can be integrated into actual IS design. For IS practitioners, the CSIS artifacts provide an easy method to design a secure system. While more research and practical studies are needed on context, the experience gained from the action research interventions in an organization indicated that the CSIS design artifact for IS design was relevant to developing secure systems in practice.

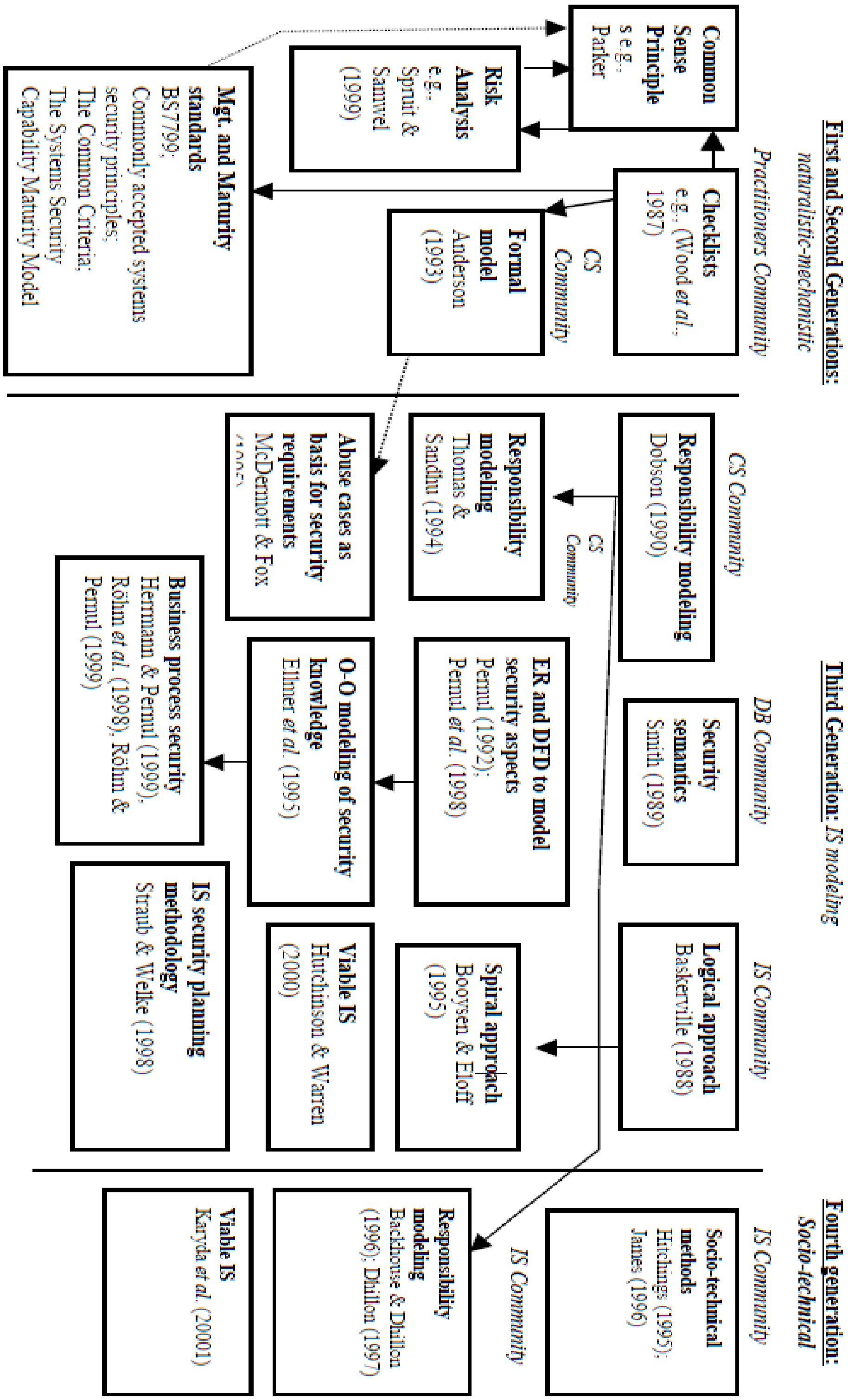
Appendix A.

Background and influences of different SIS approaches[1] (Siponen, 2002)


Interpretivism (anti-positivism)
Positivism
IS/MIS
Software engineering
Computer Science including data modeling
Mathematics
Disciplines of Information Security
IS/MIS Security
Practitioners
Computer Security
Database Security
Cryptology/ communications security
SIS Paradigms/ Approaches
Viable IS
Responsibility Approaches
Maturity criteria
Information Modeling
Formal Methods
Risk mgt.
Security-Modified IS Development Approaches
Mgt. Oriented standard
Business Process
Checklists
Security mgt. Cookbooks/ common sense principles

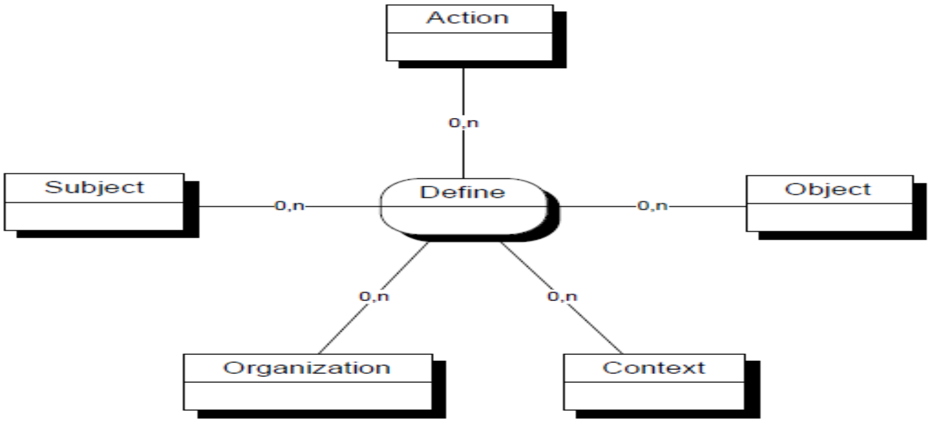
[1] The arrow shows influences. The dotted lines illustrate a weak influence.

Generations of SISD (Siponen, 2002)

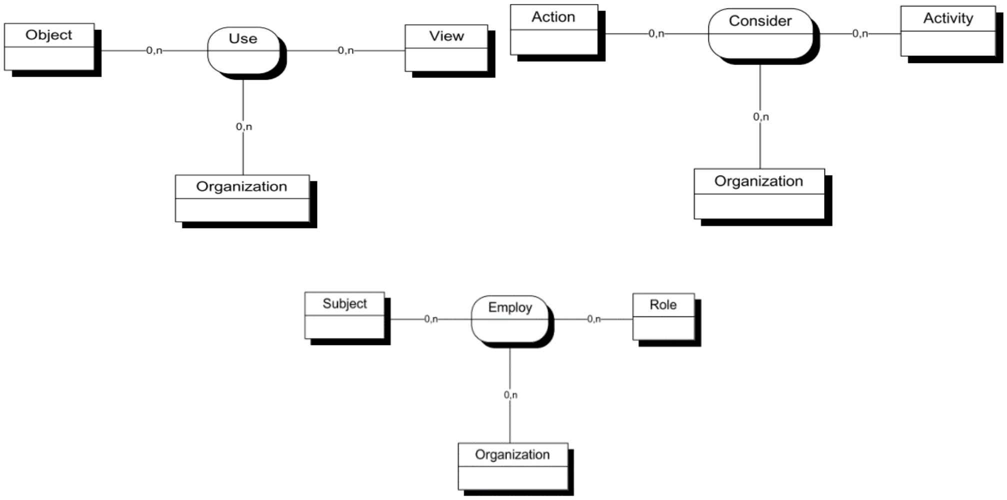


Appendix B.

Define Relationship in the OrBAC model (El Kalam et al., 2003)



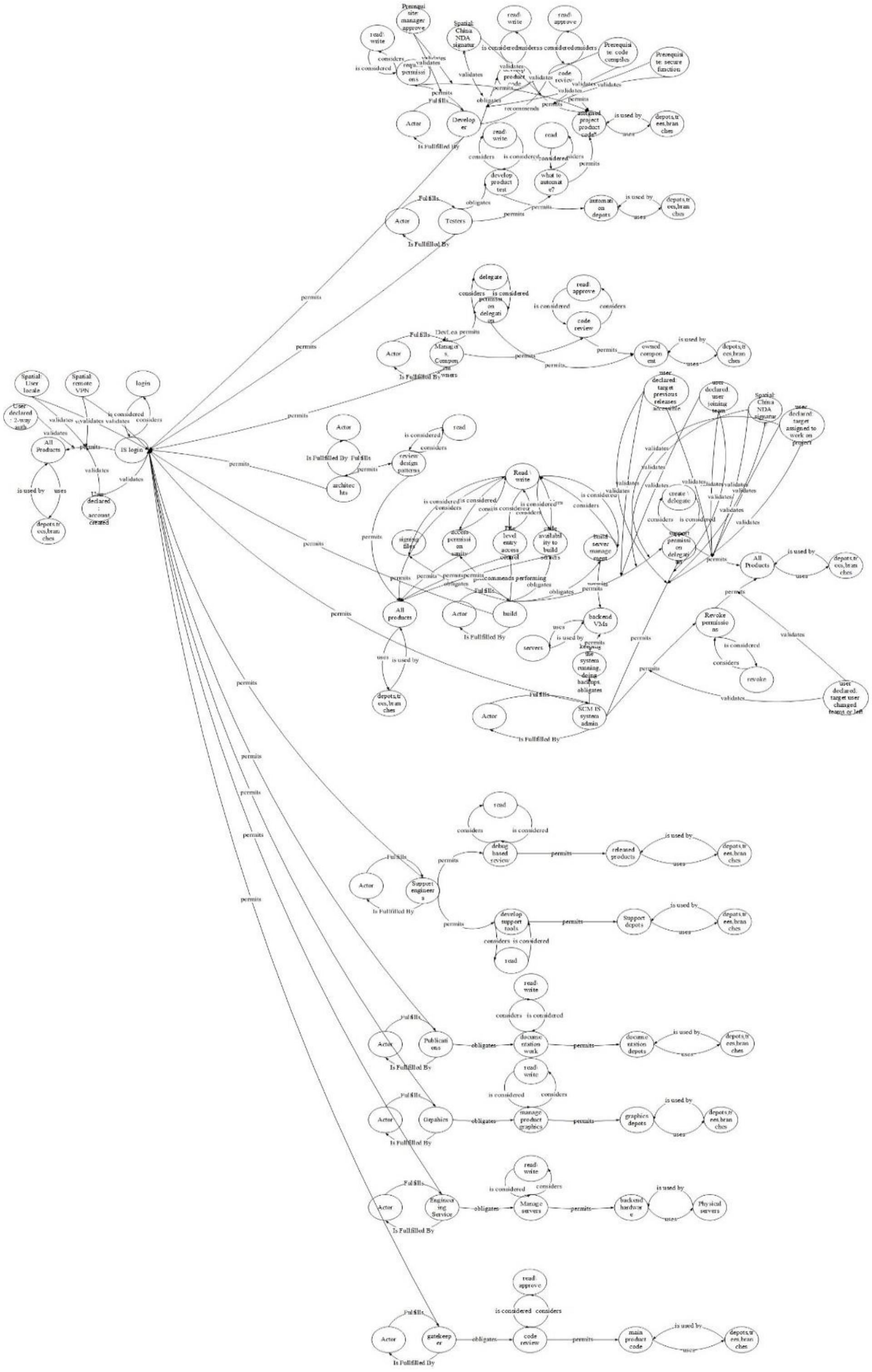
OrBAC relationships



Appendix C.

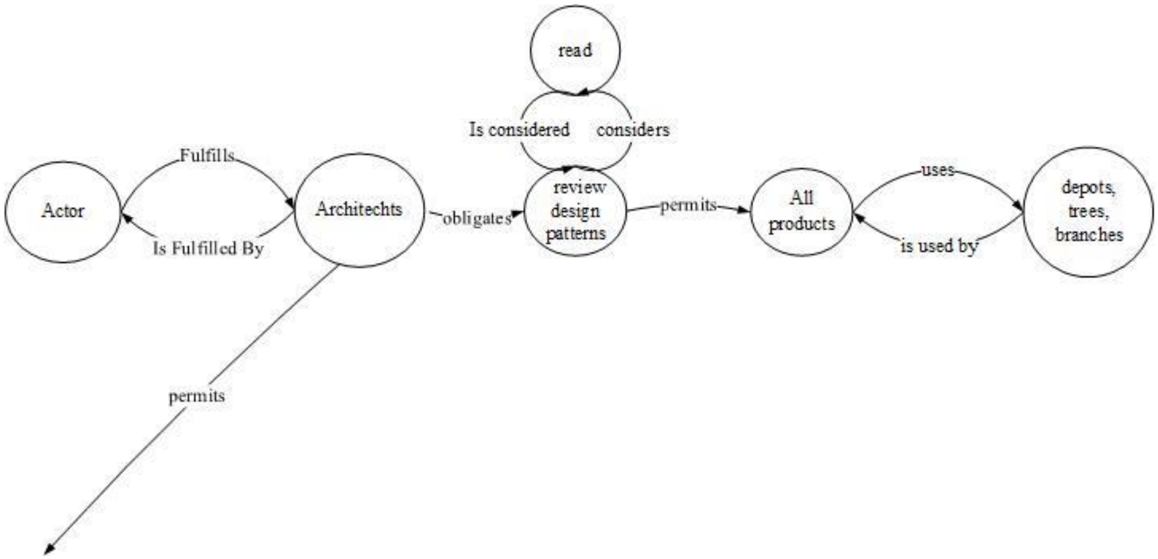
CSIS artifact for SC

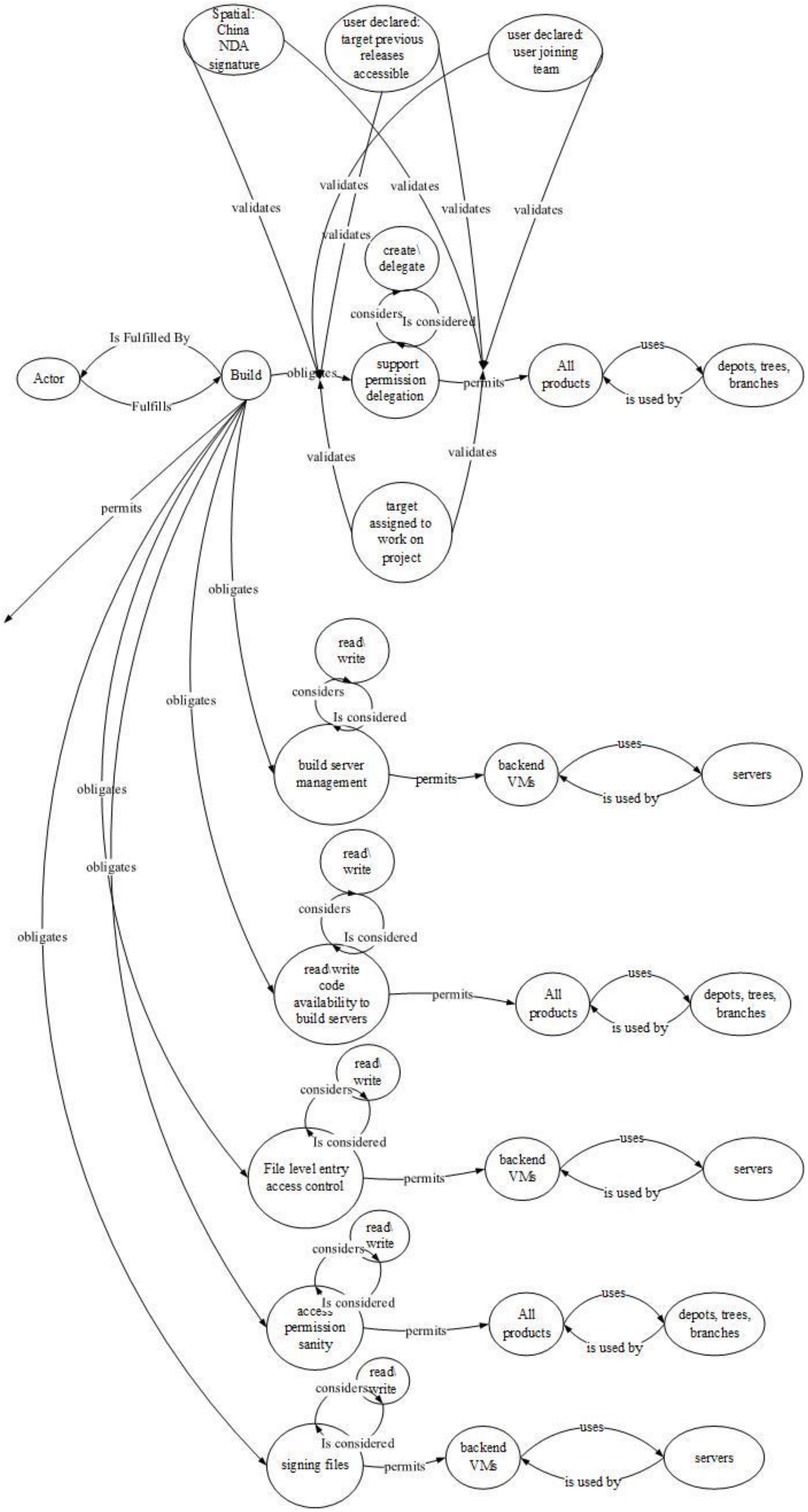
Cycle 1

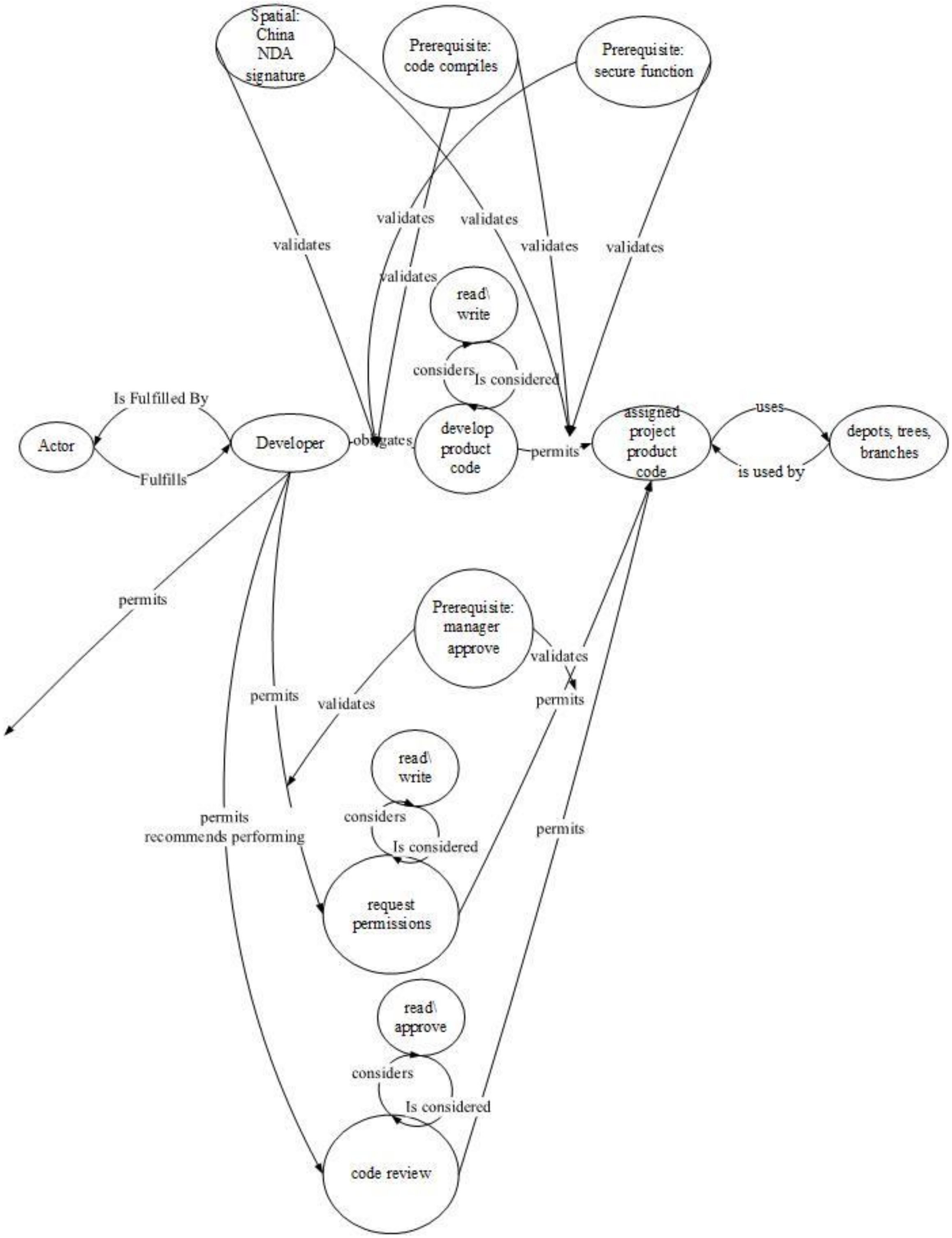


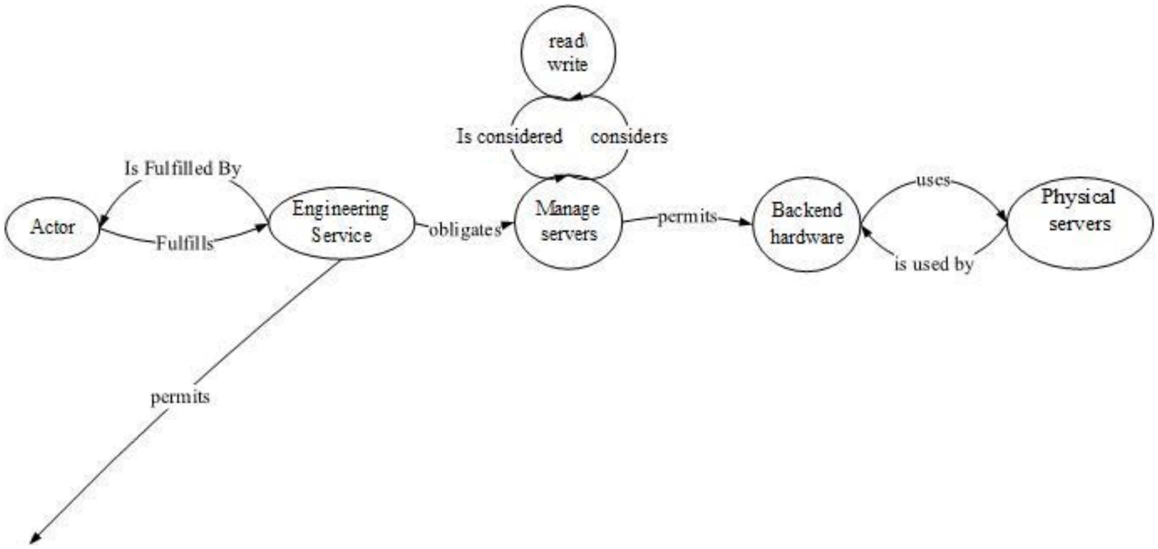
CSIS artifacts for SC

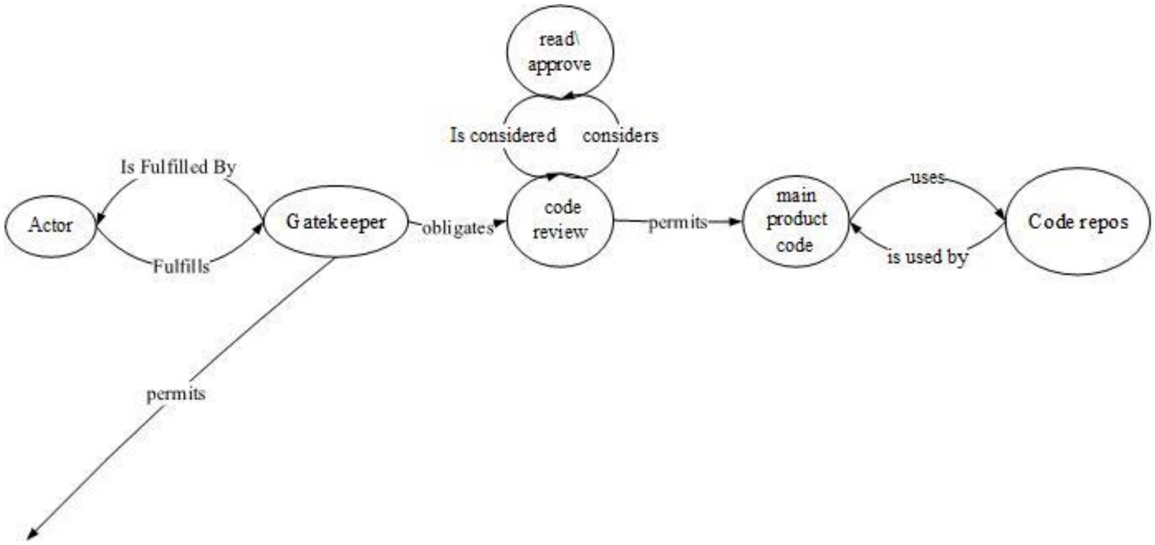
Cycle 2

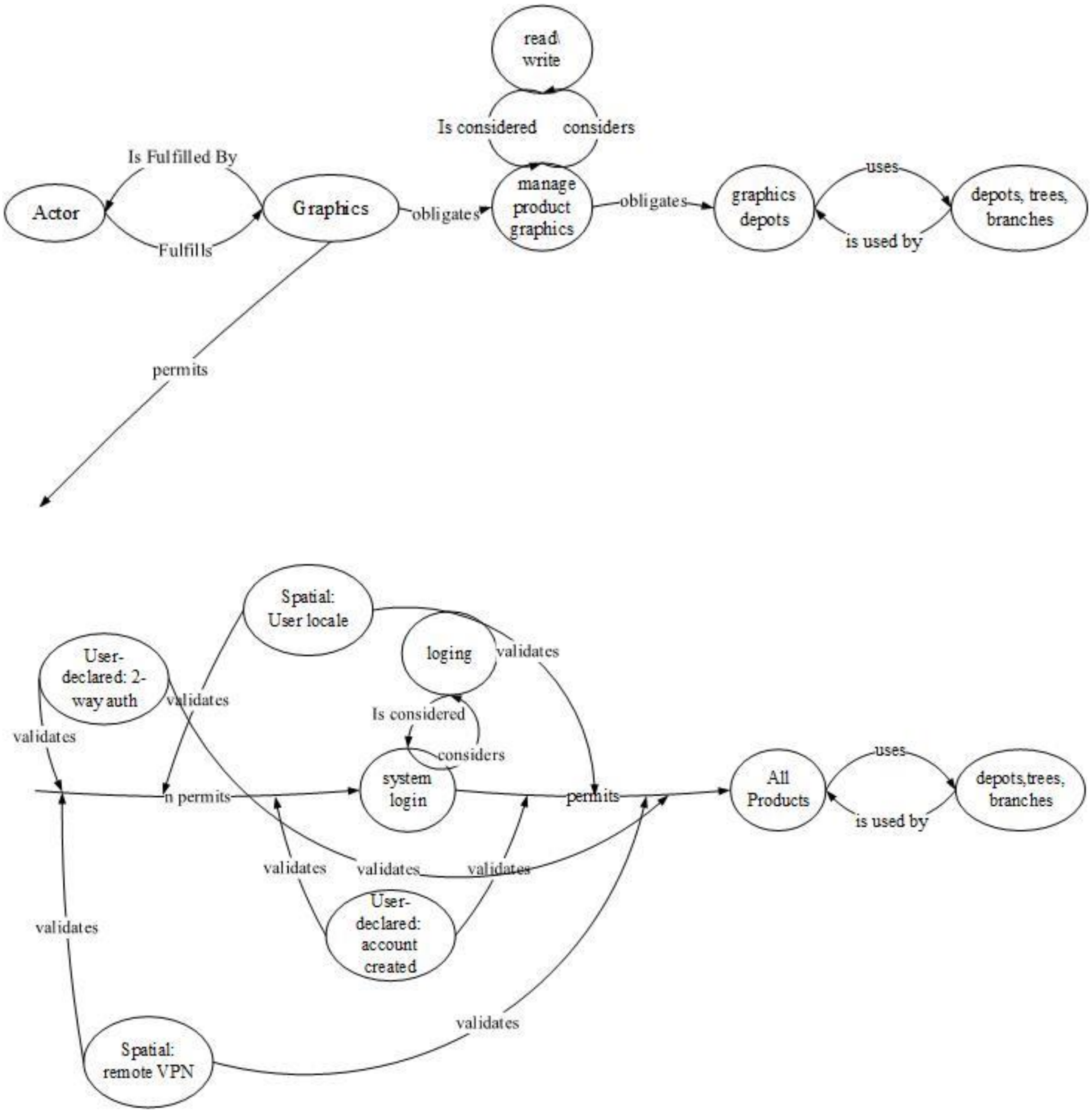


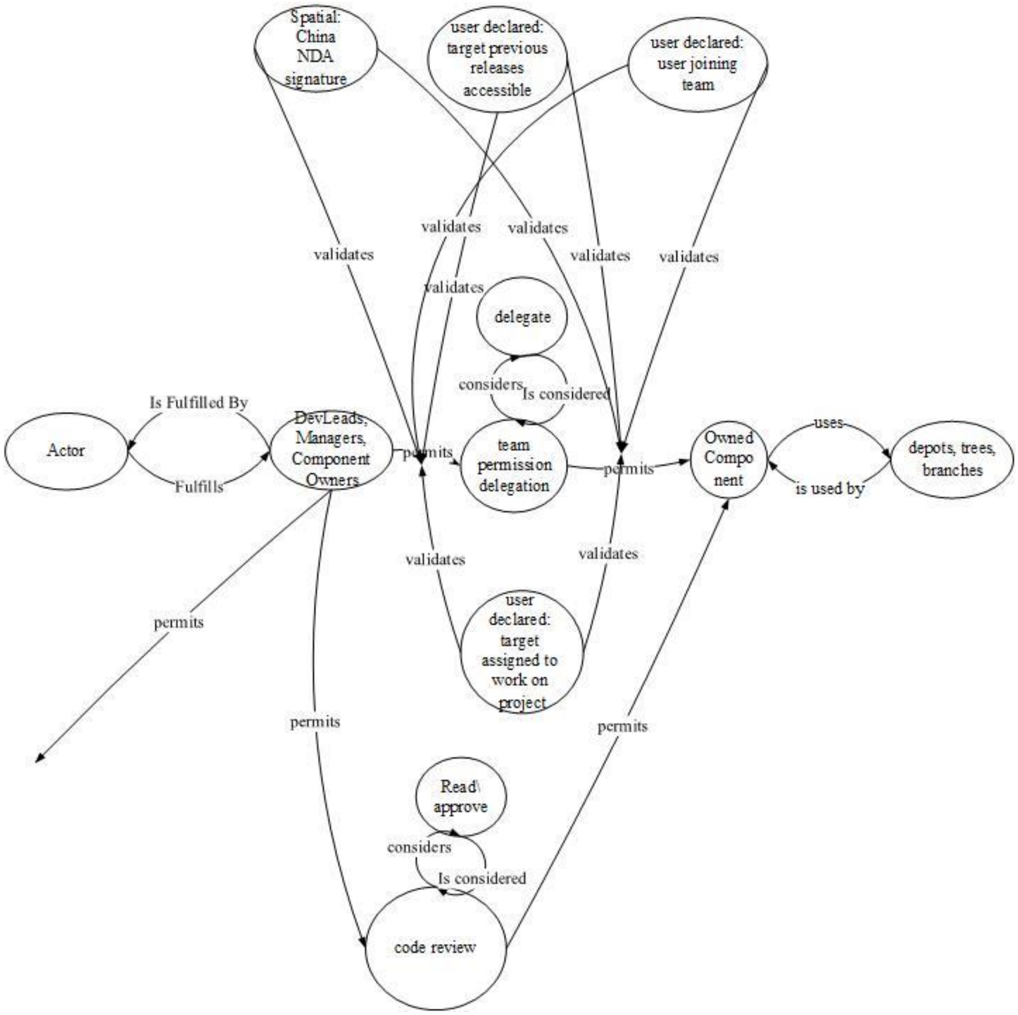

Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
create\ delegate
considers
Is considered
Is Fulfilled By
Actor
Fulfills
Build
obligates
support permission delegation
permits
All products
uses
is used by
depots, trees, branches
validates
validates
target assigned to work on project
permits
obligates
read\ write
considers
Is considered
obligates
build server management
permits
backend VMs
uses
is used by
servers
obligates
obligates
read\ write
considers
Is considered
obligates
read\write code availability to build servers
permits
All products
uses
is used by
depots, trees, branches
read\ write
considers
Is considered
File level entry access control
permits
backend VMs
uses
is used by
servers
read\ write
considers
Is considered
access permission sanity
permits
All products
uses
is used by
depots, trees, branches
read\ write
considers
Is considered
signing files
permits
backend VMs
uses
is used by
servers

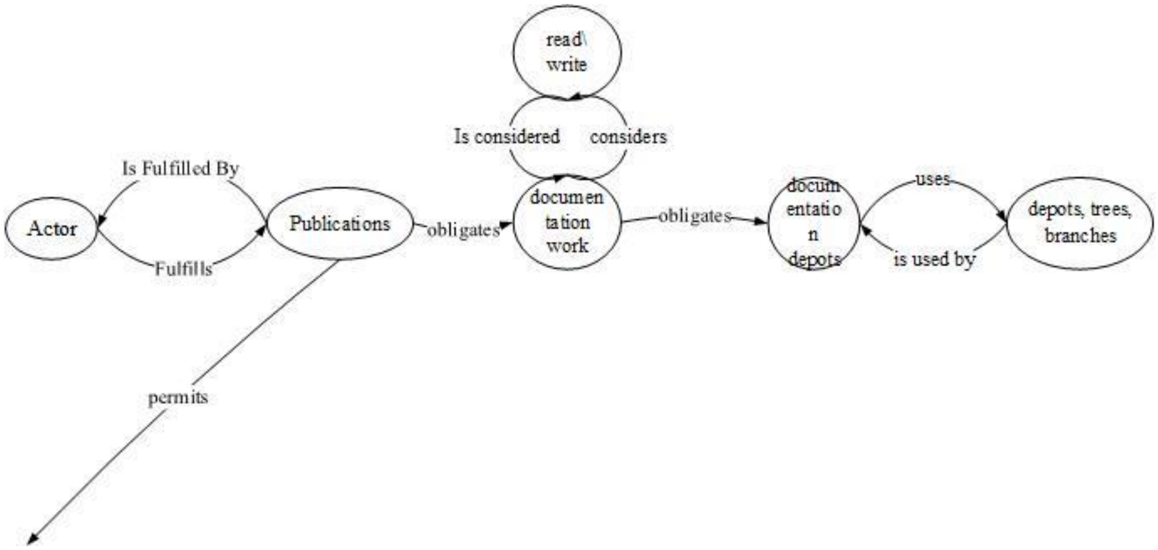

Spatial: China NDA signature
Prerequisite: code compiles
Prerequisite: secure function
validates
validates
validates
validates
validates
validates
read\ write
considers
Is considered
Actor
Is Fulfilled By
Fulfills
Developer
obligates
develop product code
permits
assigned project product code
uses
is used by
depots, trees, branches
permits
Prerequisite: manager approve
validates
permits
permits
validates
read\ write
considers
Is considered
permits
recommends performing
permits
request permissions
read\ approve
considers
Is considered
code review

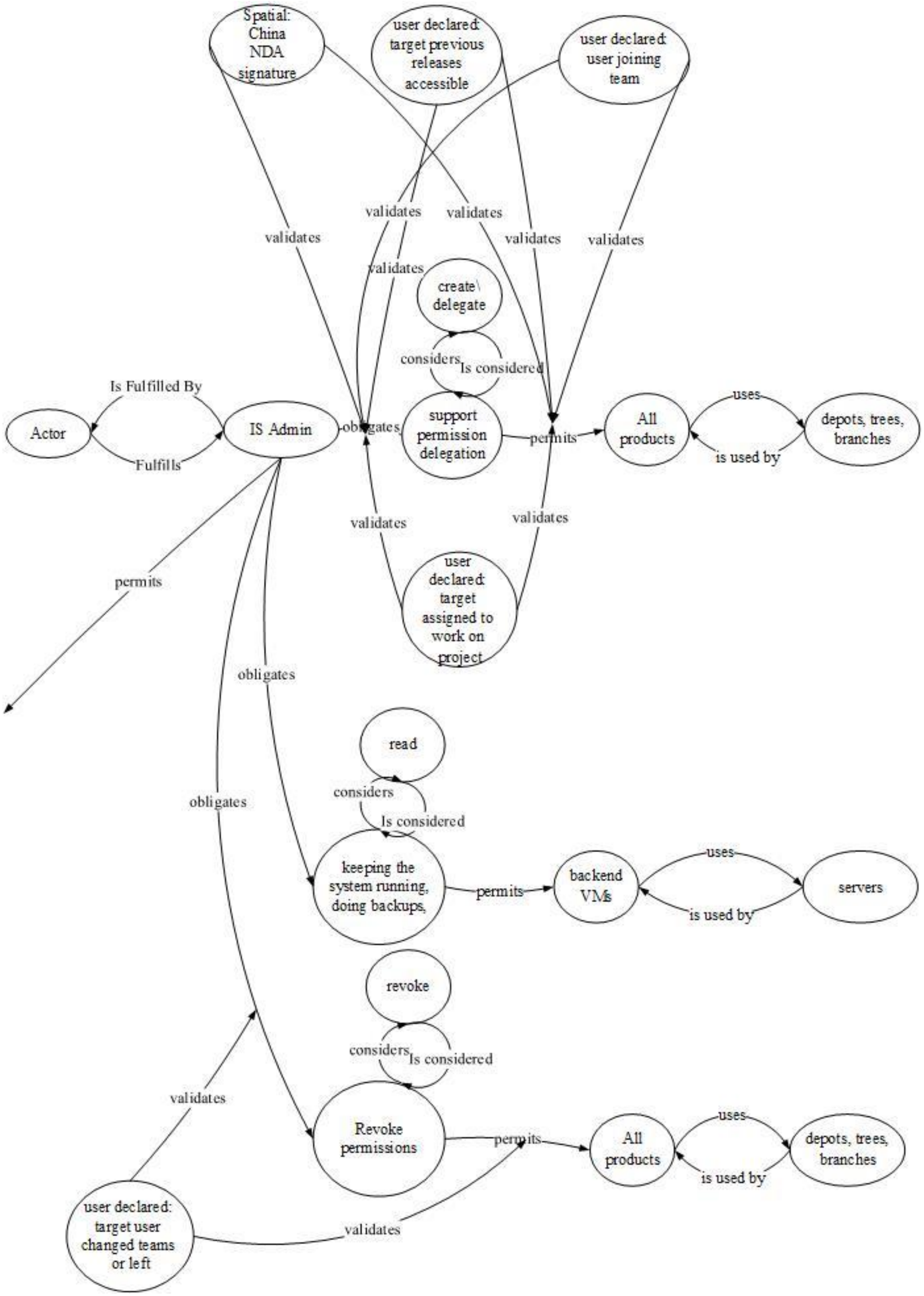

read\ write
Is considered
considers
Is Fulfilled By
Actor
Fulfills
Engineering Service
obligates
Manage servers
permits
Backend hardware
uses
is used by
Physical servers
permits

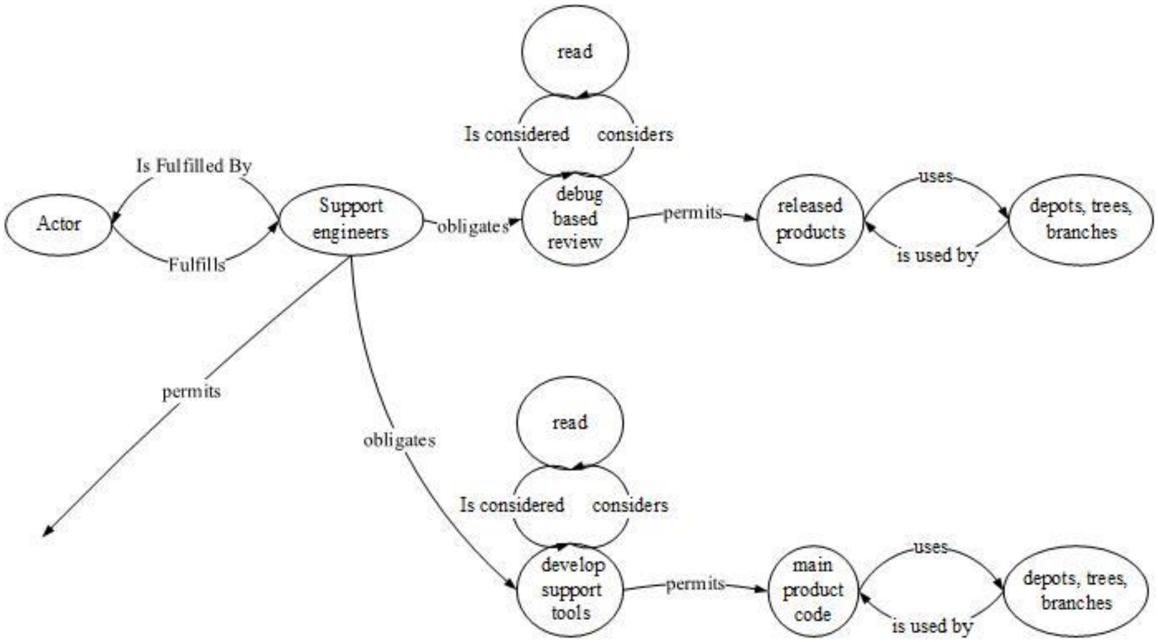

read\ approve
Is considered
considers
Is Fulfilled By
Actor
Fulfills
Gatekeeper
obligates
code review
permits
main product code
uses
is used by
Code repos
permits

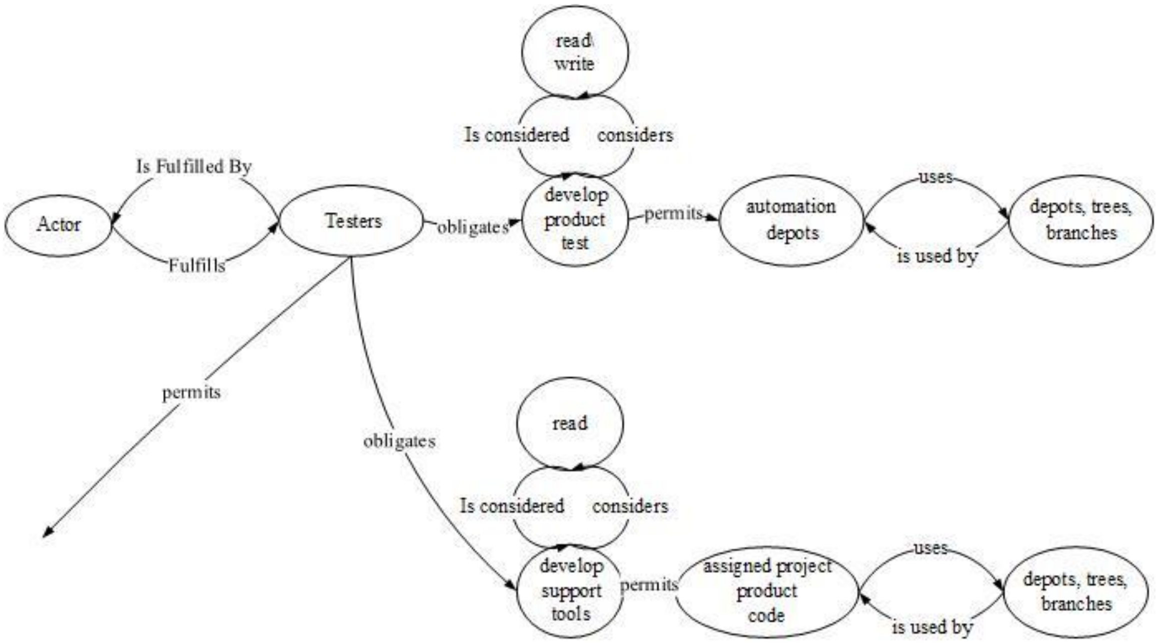

read\ write
Is considered
considers
Is Fulfilled By
Actor
Graphics
Fulfills
obligates
manage product graphics
obligates
graphics depots
uses
depots, trees, branches
is used by
permits
Spatial: User locale
User-declared: 2-way auth
loging
validates
validates
validates
Is considered
considers
n permits
system login
permits
All Products
uses
depots,trees, branches
is used by
validates
validates
validates
User-declared: account created
validates
validates
Spatial: remote VPN


Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
validates
delegate
considers
Is considered
Actor
Is Fulfilled By
Fulfills
DevLeads, Managers, Component Owners
permits
team permission delegation
permits
Owned Component
uses
is used by
depots, trees, branches
validates
validates
user declared: target assigned to work on project
permits
permits
permits
Read\ approve
considers
Is considered
code review


read\ write
Is considered
considers
Is Fulfilled By
Actor
Fulfills
Publications
obligates
documentation work
obligates
documentation depots
uses
is used by
depots, trees, branches
permits

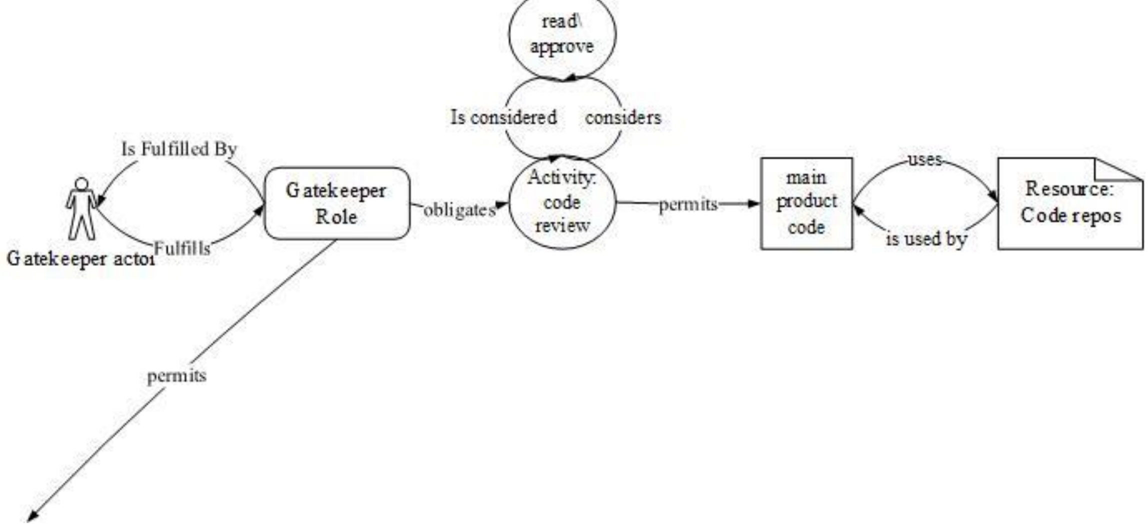

Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
create\ delegate
considers
Is considered
Actor
Is Fulfilled By
Fulfills
IS Admin
obligates
support permission delegation
permits
All products
uses
is used by
depots, trees, branches
validates
validates
user declared: target assigned to work on project
permits
obligates
obligates
read
considers
Is considered
keeping the system running, doing backups,
permits
backend VMs
uses
is used by
servers
revoke
considers
Is considered
validates
Revoke permissions
permits
All products
uses
is used by
depots, trees, branches
user declared: target user changed teams or left
validates

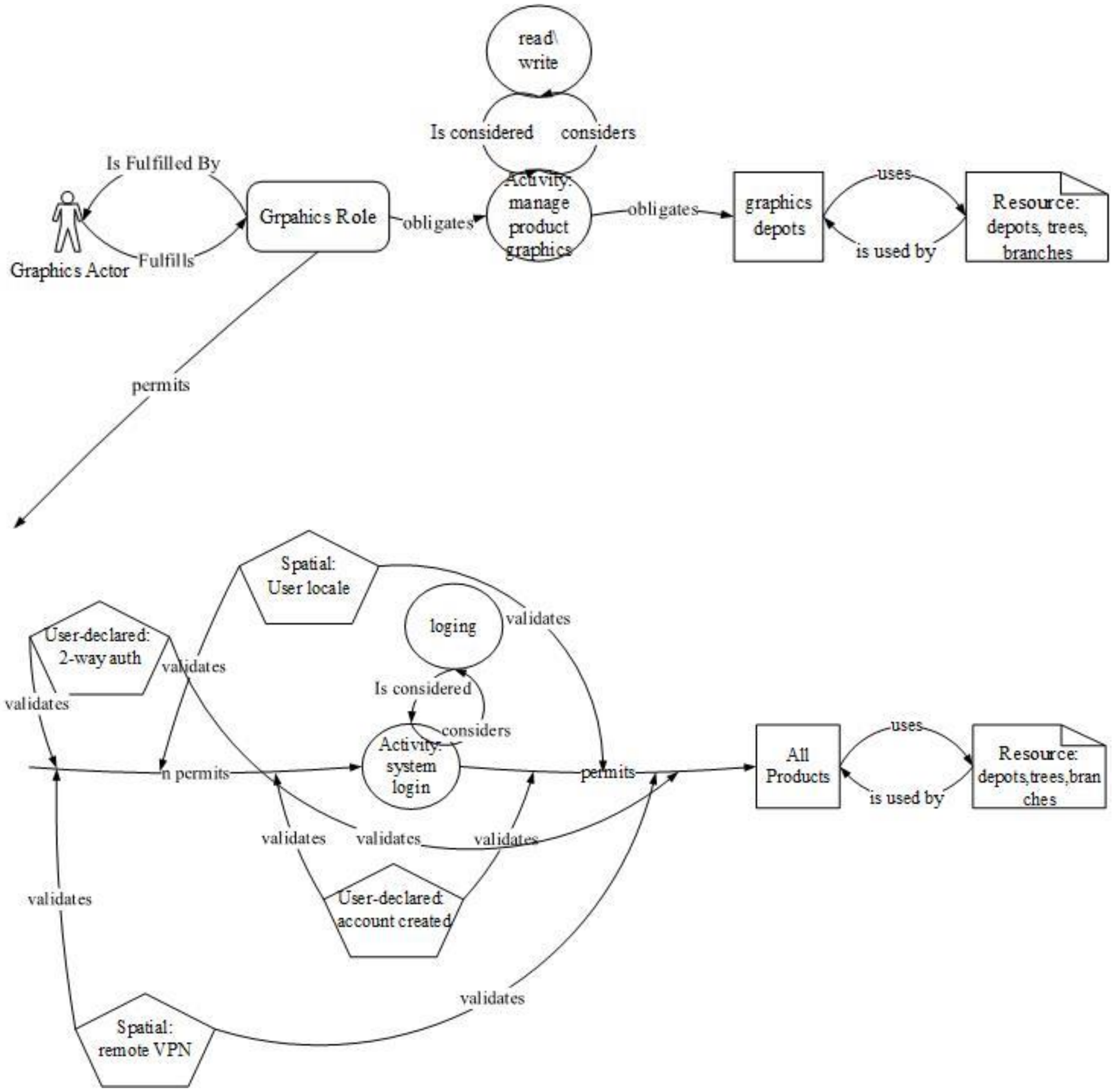

read
Is considered
considers
Is Fulfilled By
Actor
Support engineers
obligates
debug based review
permits
released products
uses
is used by
depots, trees, branches
Fulfills
permits
obligates
read
Is considered
considers
develop support tools
permits
main product code
uses
is used by
depots, trees, branches

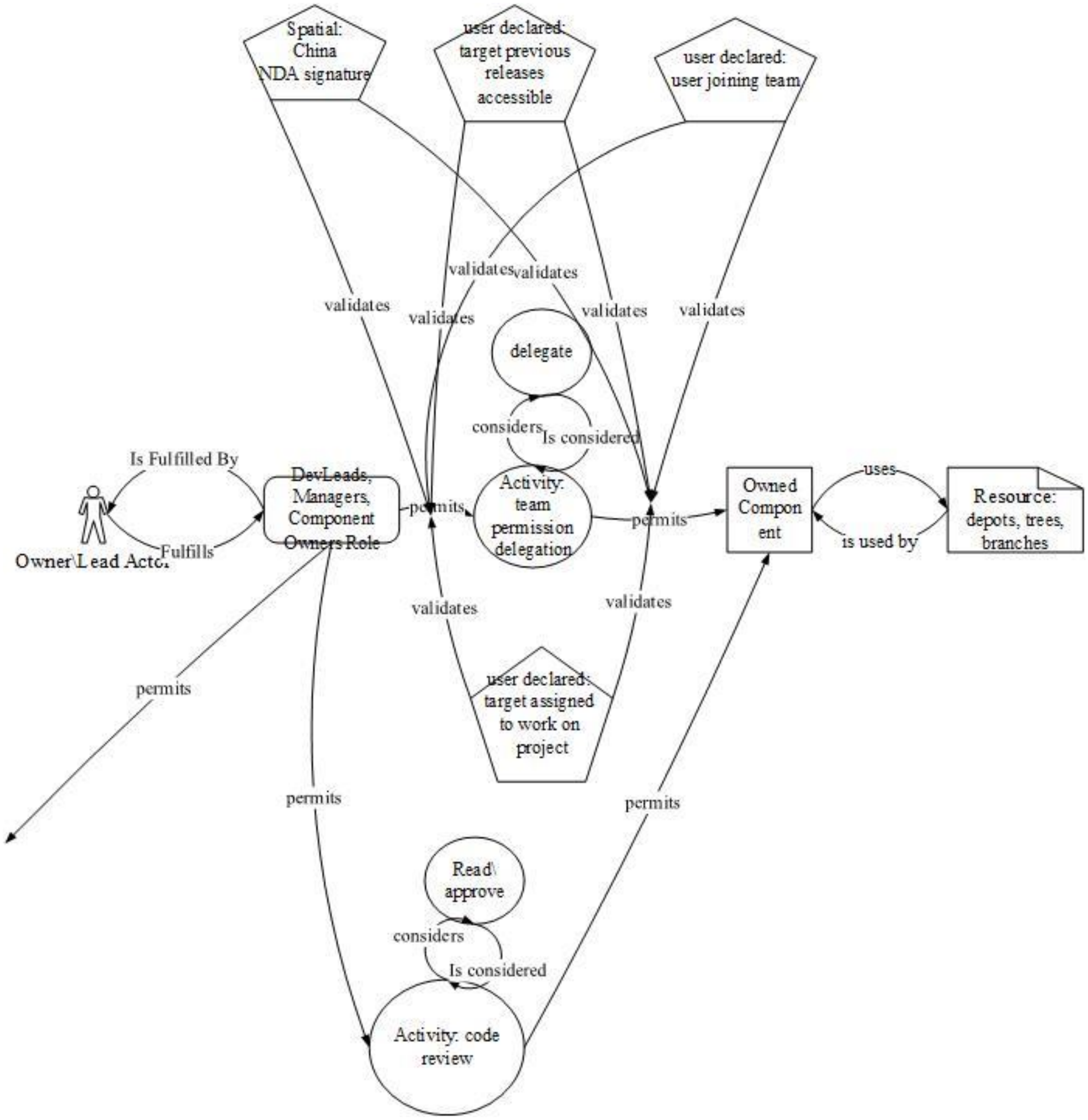


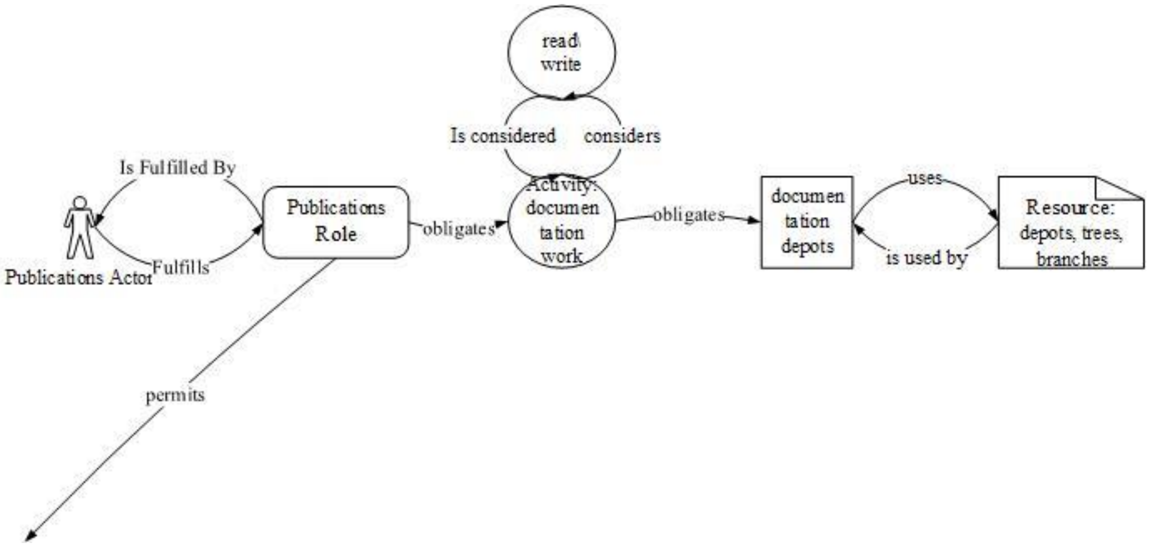
CSIS artifacts for SC

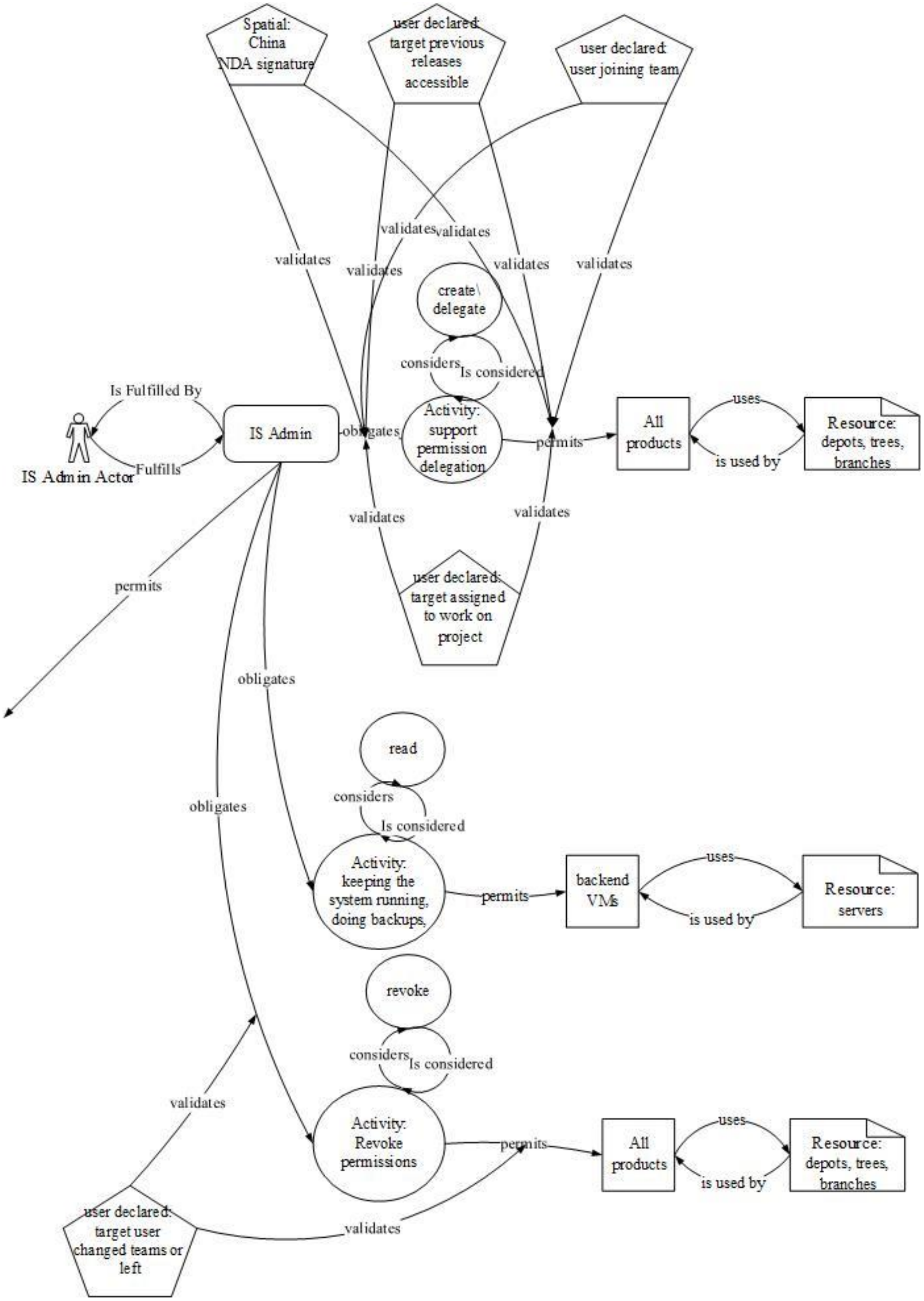
Cycle 3

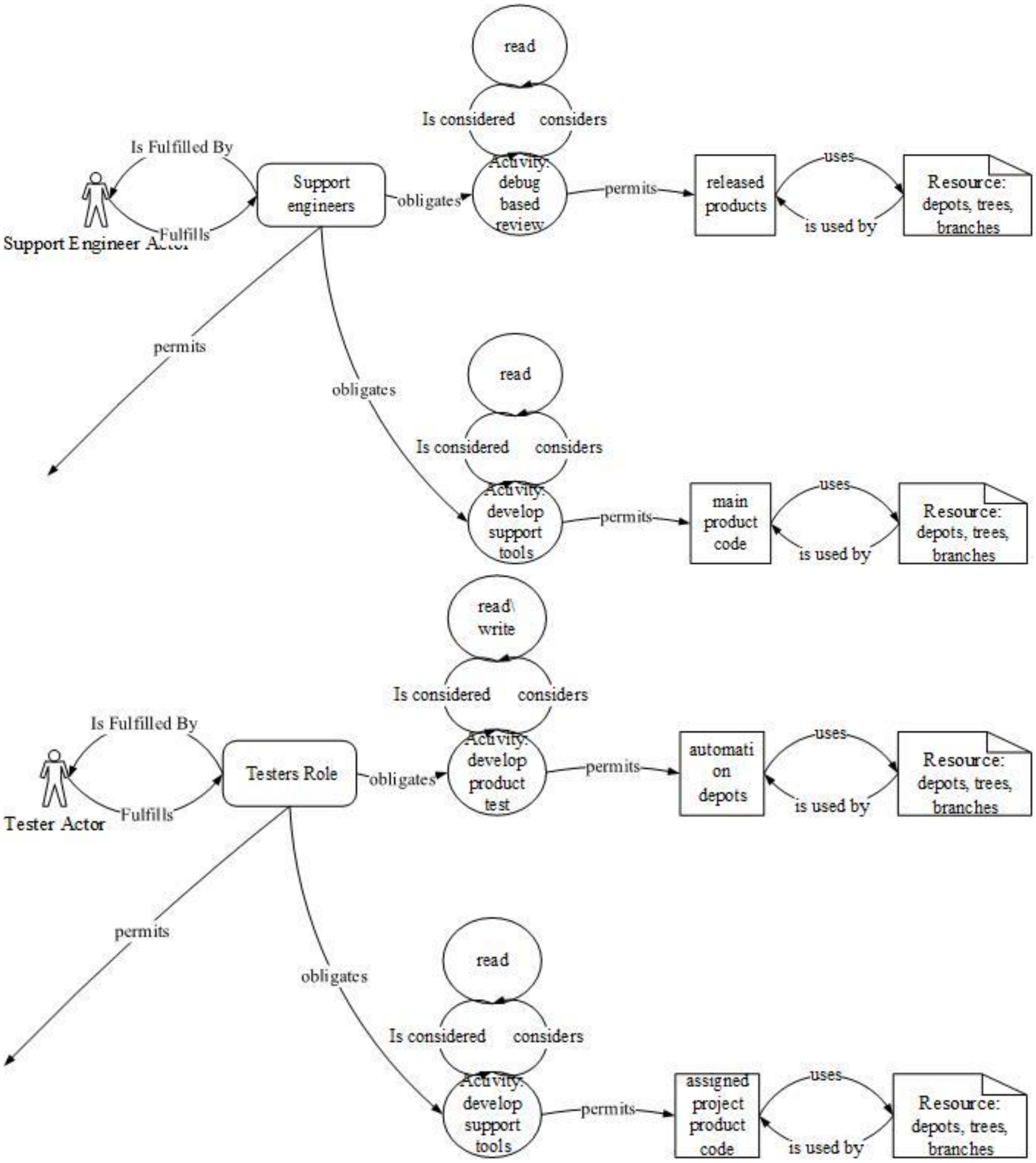


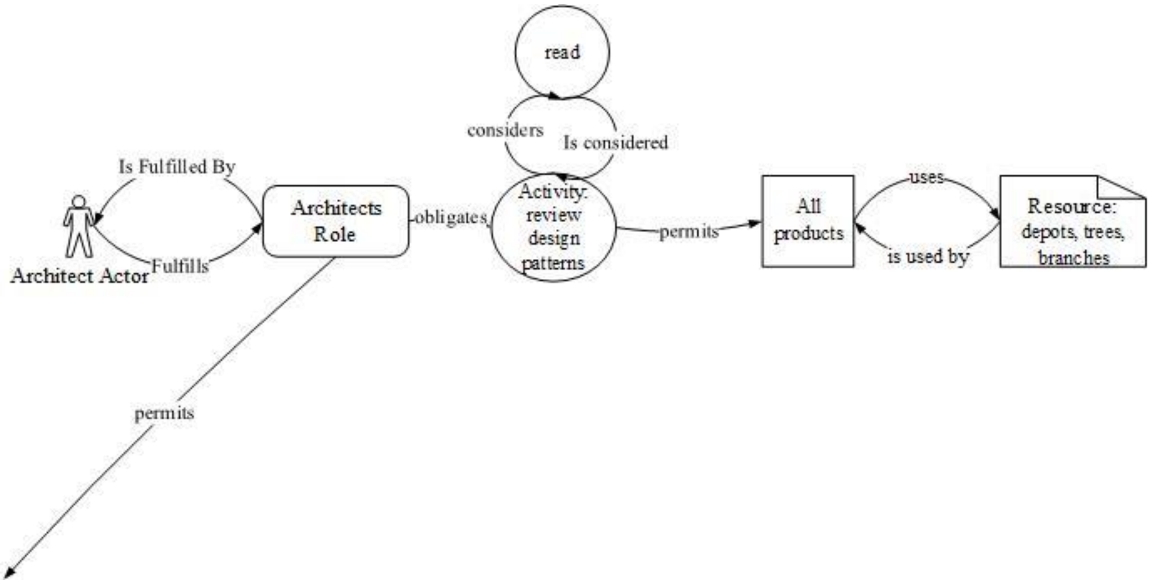

read\
write
Is considered
considers
Is Fulfilled By
Grpahics Role
obligates
Activity:
manage
product
graphics
obligates
graphics
depots
uses
Resource:
depots, trees,
branches
is used by
Graphics Actor
Fulfills
permits
Spatial:
User locale
validates
User-declared:
2-way auth
validates
validates
loging
Is considered
considers
Activity:
system
login
n permits
permits
All
Products
uses
Resource:
depots,trees,bran
ches
is used by
validates
validates
validates
User-declared:
account created
validates
validates
Spatial:
remote VPN

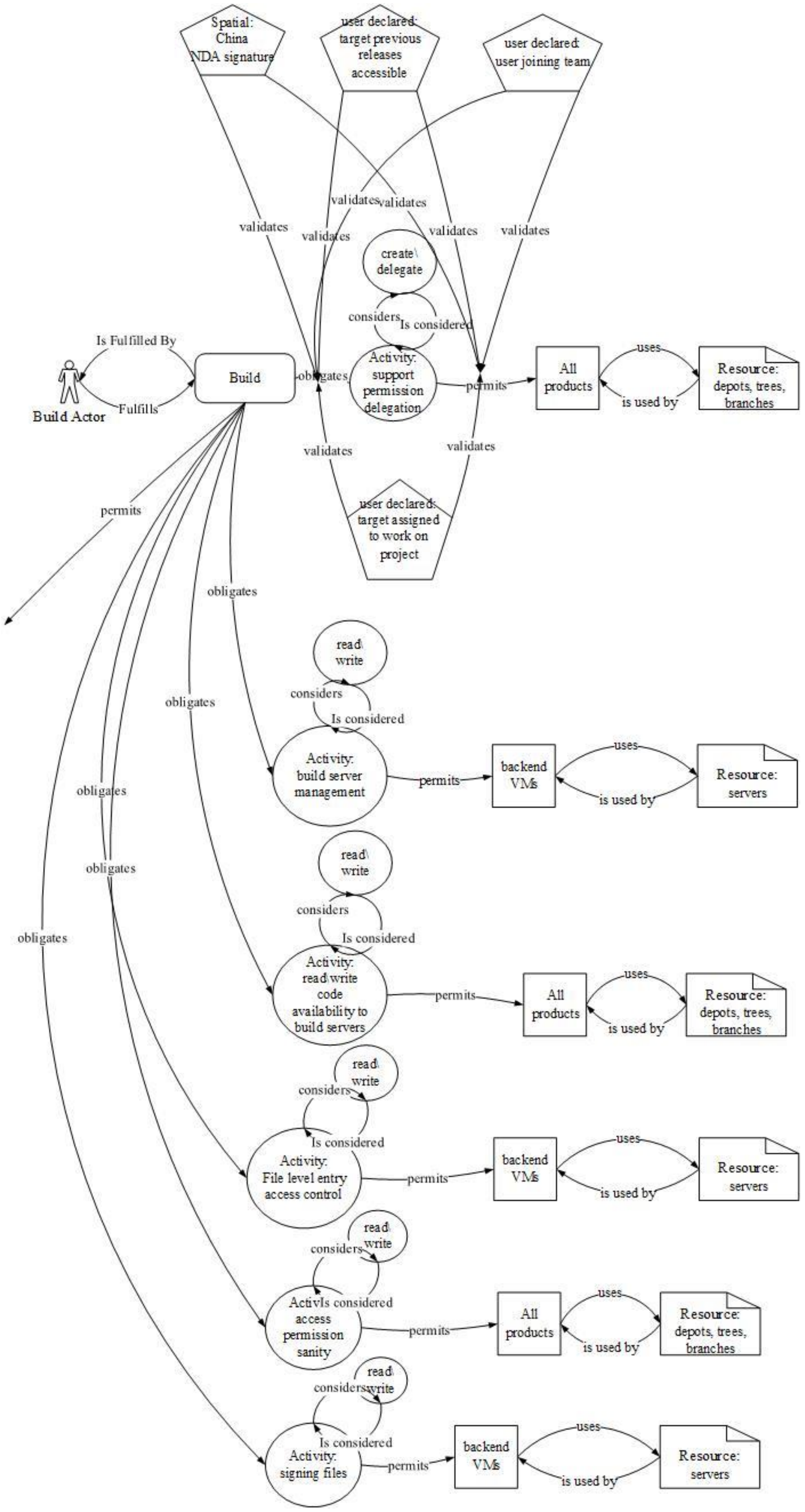

Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
delegate
considers
Is considered
Is Fulfilled By
Fulfills
Owner\Lead Actor
DevLeads, Managers, Component Owners Role
permits
Activity: team permission delegation
permits
Owned Component
uses
is used by
Resource: depots, trees, branches
validates
validates
user declared: target assigned to work on project
permits
permits
permits
Read\ approve
considers
Is considered
Activity: code review

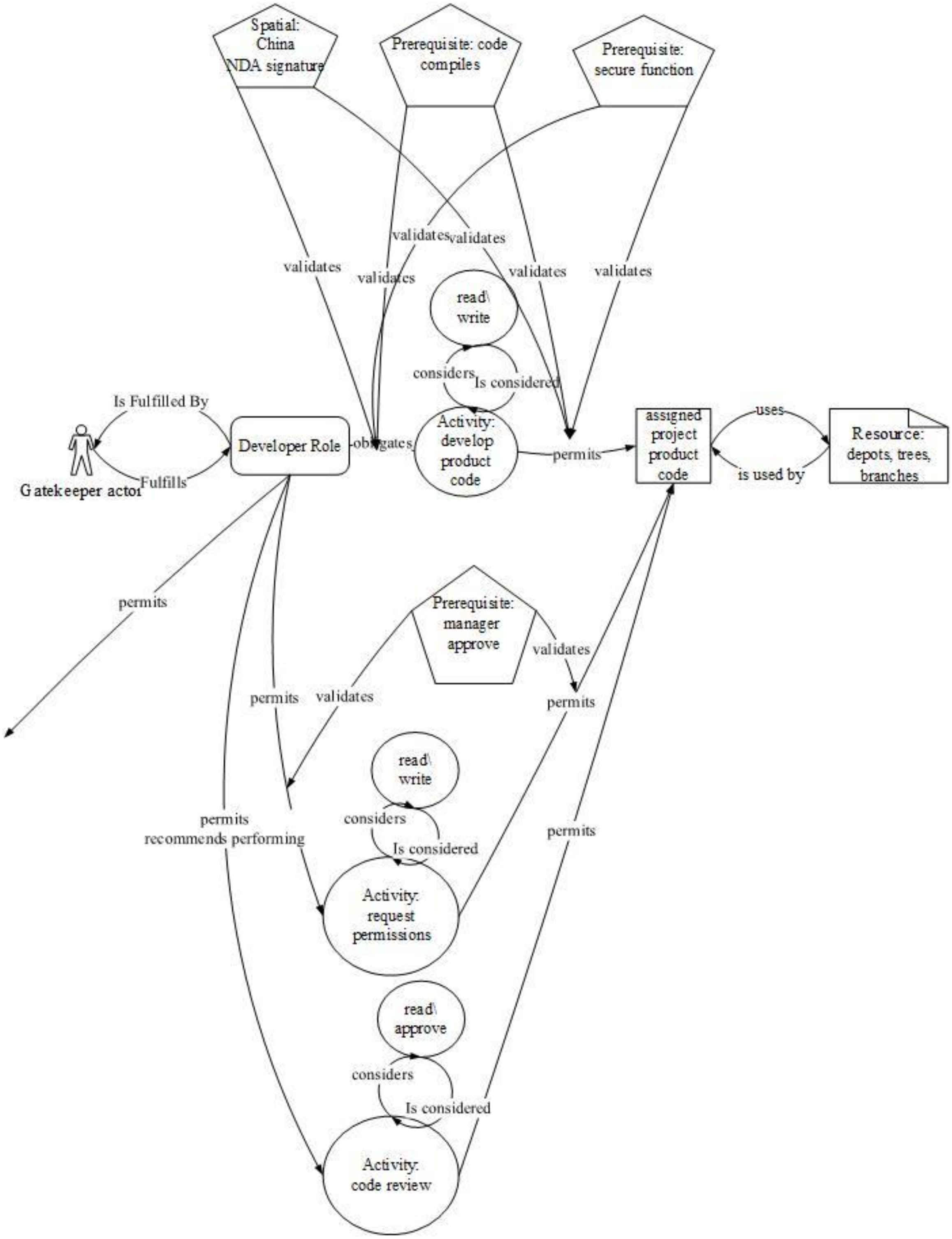

read\
write
Is considered
considers
Is Fulfilled By
Publications Actor
Fulfills
Publications
Role
obligates
Activity:
documen
tation
work
obligates
documen
tation
depots
uses
is used by
Resource:
depots, trees,
branches
permits

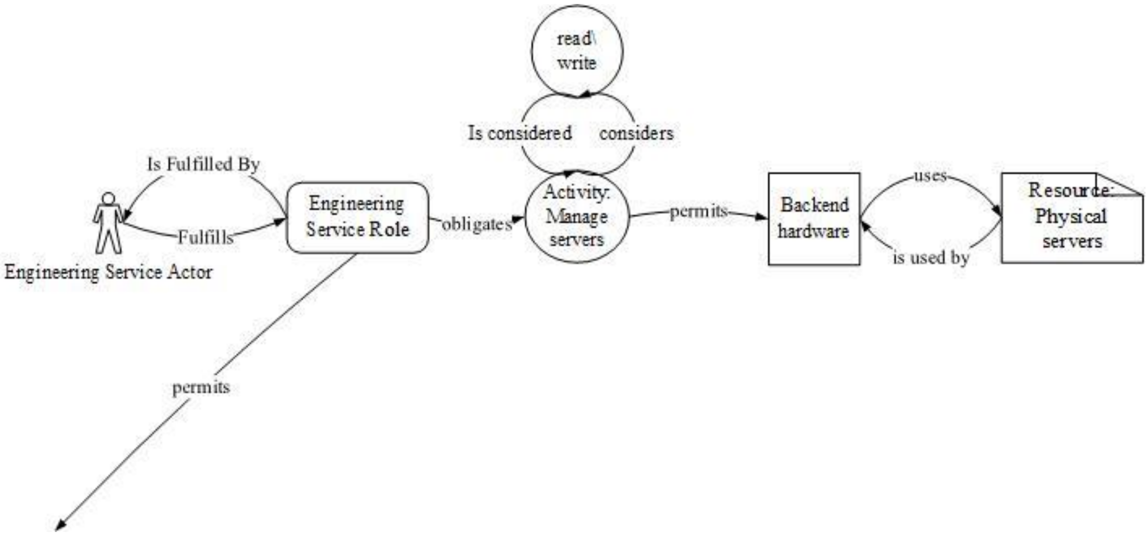

Spatial:
China
NDA signature
user declared:
target previous
releases
accessible
user declared:
user joining team
validates
validates
validates
validates
validates
validates
create\
delegate
considers
Is considered
Is Fulfilled By
IS Admin
obligates
Activity:
support
permission
delegation
permits
All
products
uses
is used by
Resource:
depots, trees,
branches
IS Admin Actor
Fulfills
validates
validates
user declared:
target assigned
to work on
project
permits
obligates
obligates
read
considers
Is considered
Activity:
keeping the
system running,
doing backups,
permits
backend
VMs
uses
is used by
Resource:
servers
revoke
considers
Is considered
validates
Activity:
Revoke
permissions
permits
All
products
uses
is used by
Resource:
depots, trees,
branches
user declared:
target user
changed teams or
left
validates


read
Is considered
considers
Is Fulfilled By
Support engineers
obligates
Activity: debug based review
permits
released products
uses
is used by
Resource: depots, trees, branches
Support Engineer Actor
Fulfills
permits
obligates
read
Is considered
considers
Activity: develop support tools
permits
main product code
uses
is used by
Resource: depots, trees, branches
read\ write
Is considered
considers
Is Fulfilled By
Testers Role
obligates
Activity: develop product test
permits
automation depots
uses
is used by
Resource: depots, trees, branches
Tester Actor
Fulfills
permits
obligates
read
Is considered
considers
Activity: develop support tools
permits
assigned project product code
uses
is used by
Resource: depots, trees, branches


read
considers
Is considered
Is Fulfilled By
Architects Role
obligates
Activity: review design patterns
permits
All products
uses
is used by
Resource: depots, trees, branches
Architect Actor
Fulfills
permits

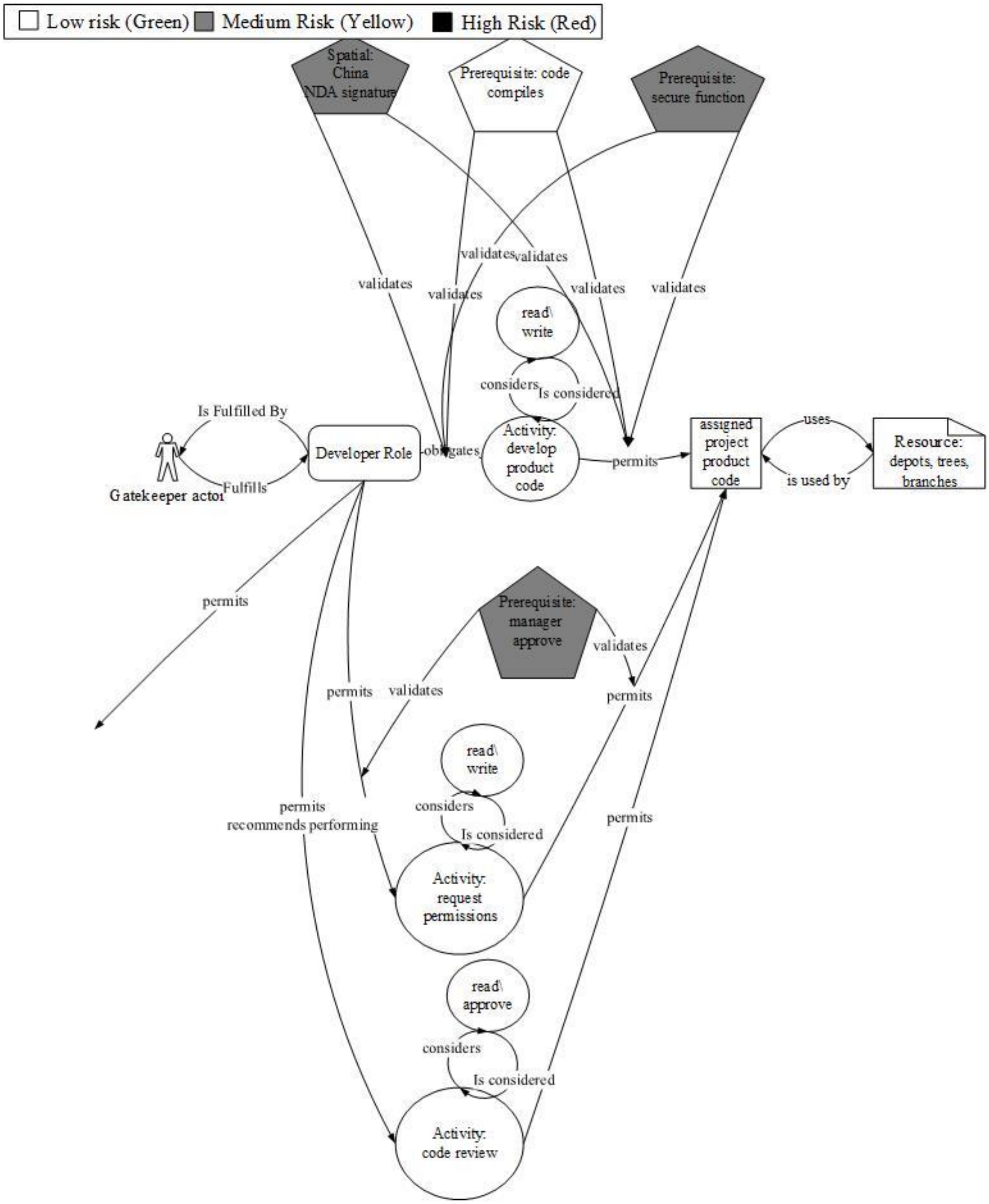

Spatial:
China
NDA signature
user declared:
target previous
releases
accessible
user declared:
user joining team
validates
validates
validates
validates
validates
validates
create\
delegate
considers
Is considered
Is Fulfilled By
Build
obligates
Activity:
support
permission
delegation
permits
All
products
uses
is used by
Resource:
depots, trees,
branches
Build Actor
Fulfills
validates
validates
permits
user declared:
target assigned
to work on
project
obligates
read\
write
considers
Is considered
obligates
Activity:
build server
management
permits
backend
VMs
uses
is used by
Resource:
servers
obligates
obligates
read\
write
considers
Is considered
obligates
Activity:
read\write
code
availability to
build servers
permits
All
products
uses
is used by
Resource:
depots, trees,
branches
read\
write
considers
Is considered
Activity:
File level entry
access control
permits
backend
VMs
uses
is used by
Resource:
servers
read\
write
considers
ActivIs considered
access
permission
sanity
permits
All
products
uses
is used by
Resource:
depots, trees,
branches
read\
write
considers
Is considered
Activity:
signing files
permits
backend
VMs
uses
is used by
Resource:
servers

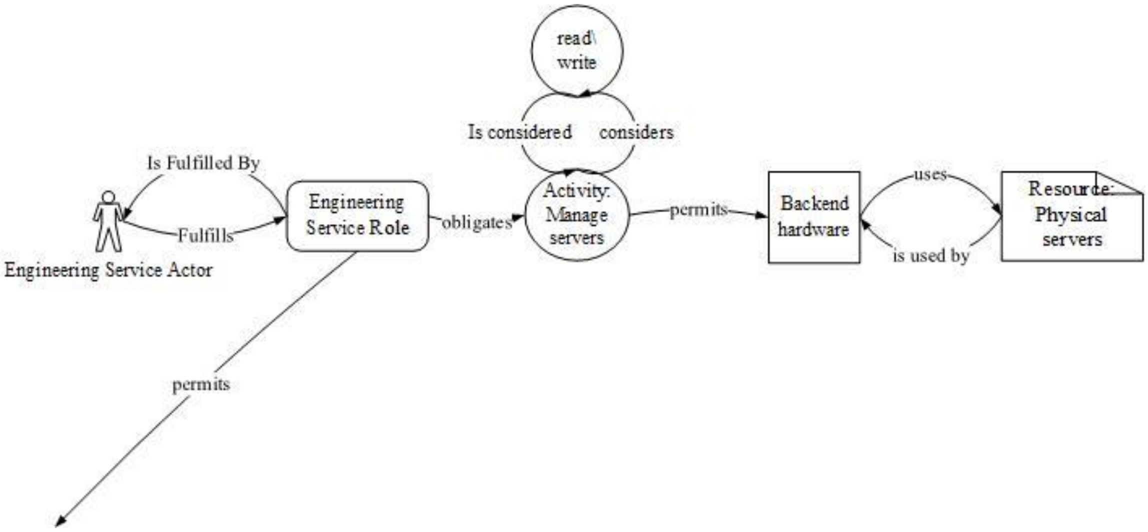

Spatial:
China
NDA signature
Prerequisite: code compiles
Prerequisite: secure function
validates
validates
validates
validates
validates
validates
validates
read\
write
considers
Is considered
Is Fulfilled By
Developer Role
obligates
Activity:
develop
product
code
permits
assigned
project
product
code
uses
is used by
Resource:
depots, trees,
branches
Gatekeeper actor
Fulfills
permits
Prerequisite:
manager
approve
validates
permits
validates
permits
read\
write
considers
Is considered
permits
recommends performing
permits
Activity:
request
permissions
read\
approve
considers
Is considered
Activity:
code review

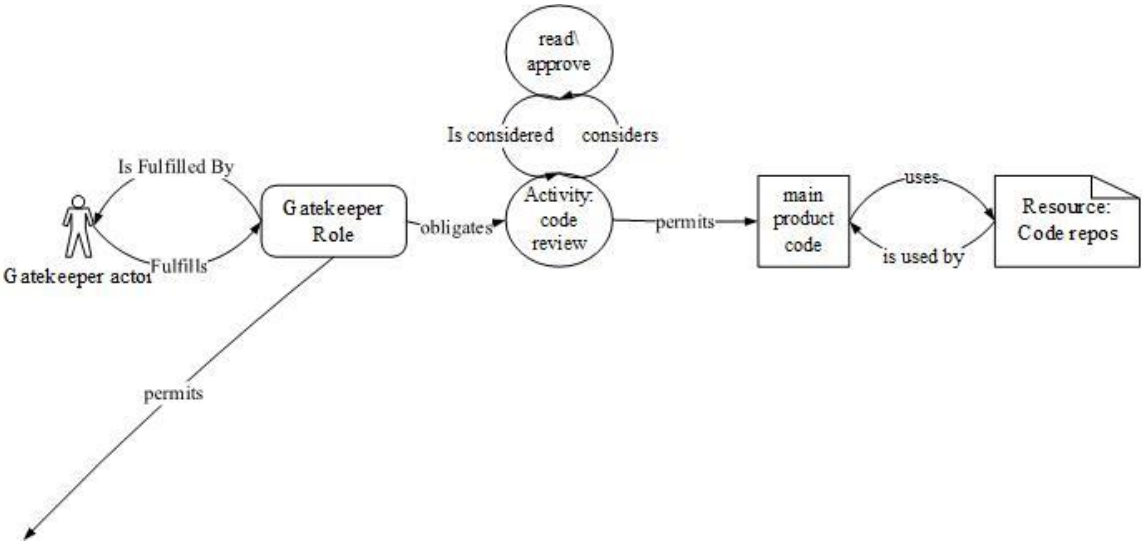

read\ write
Is considered
considers
Is Fulfilled By
Fulfills
Engineering Service Actor
Engineering Service Role
obligates
Activity: Manage servers
permits
Backend hardware
uses
is used by
Resource: Physical servers
permits

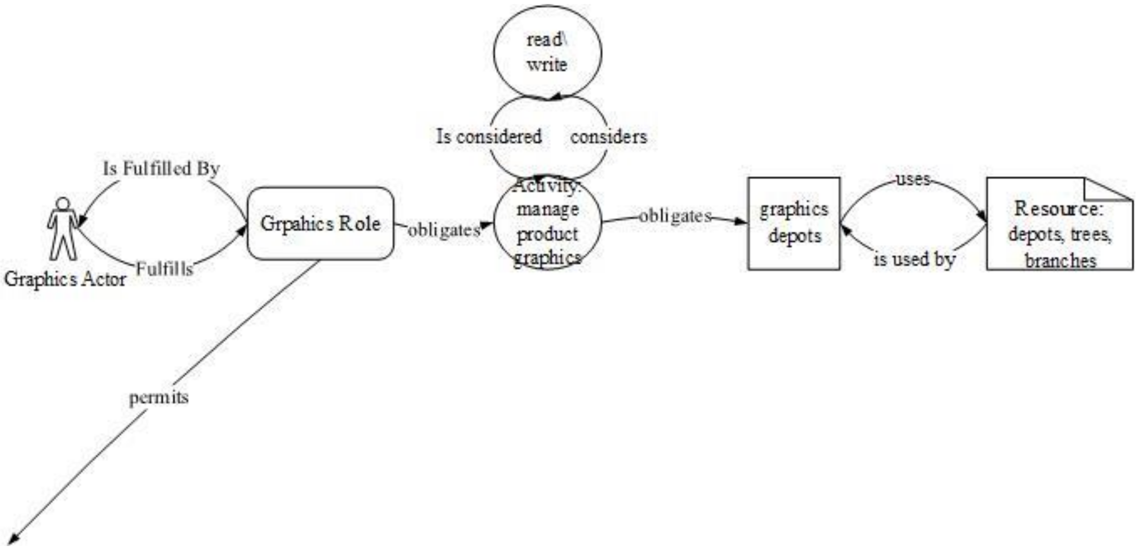
CSIS artifacts for SC

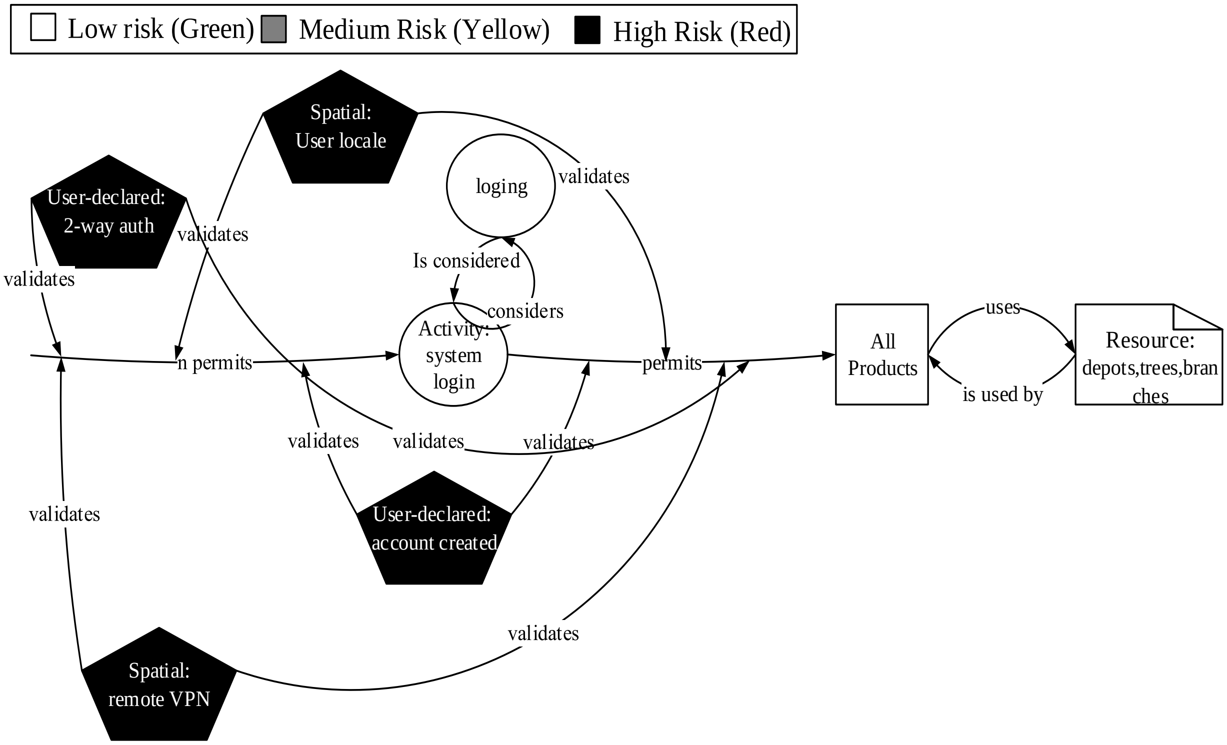
Cycle 4

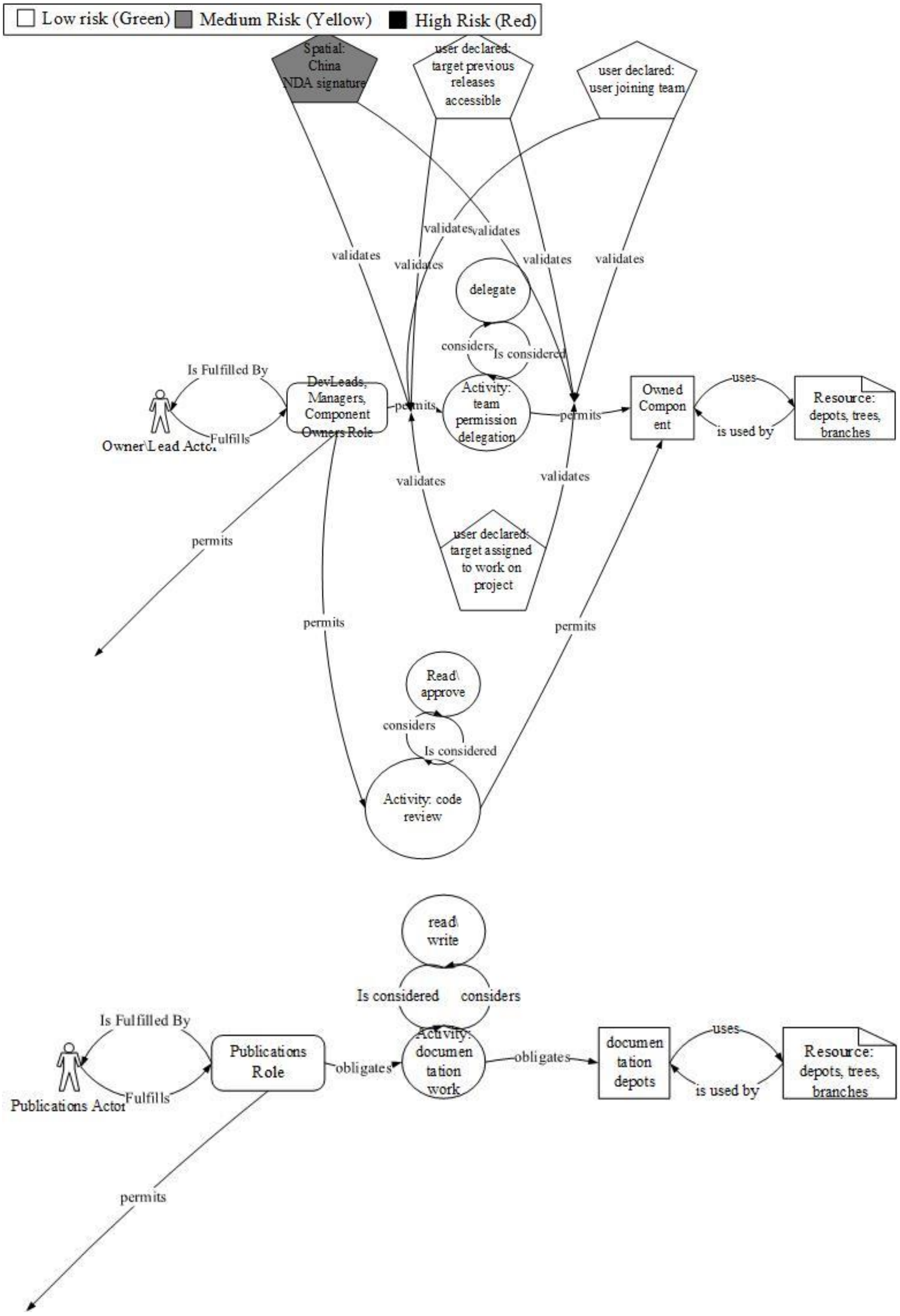


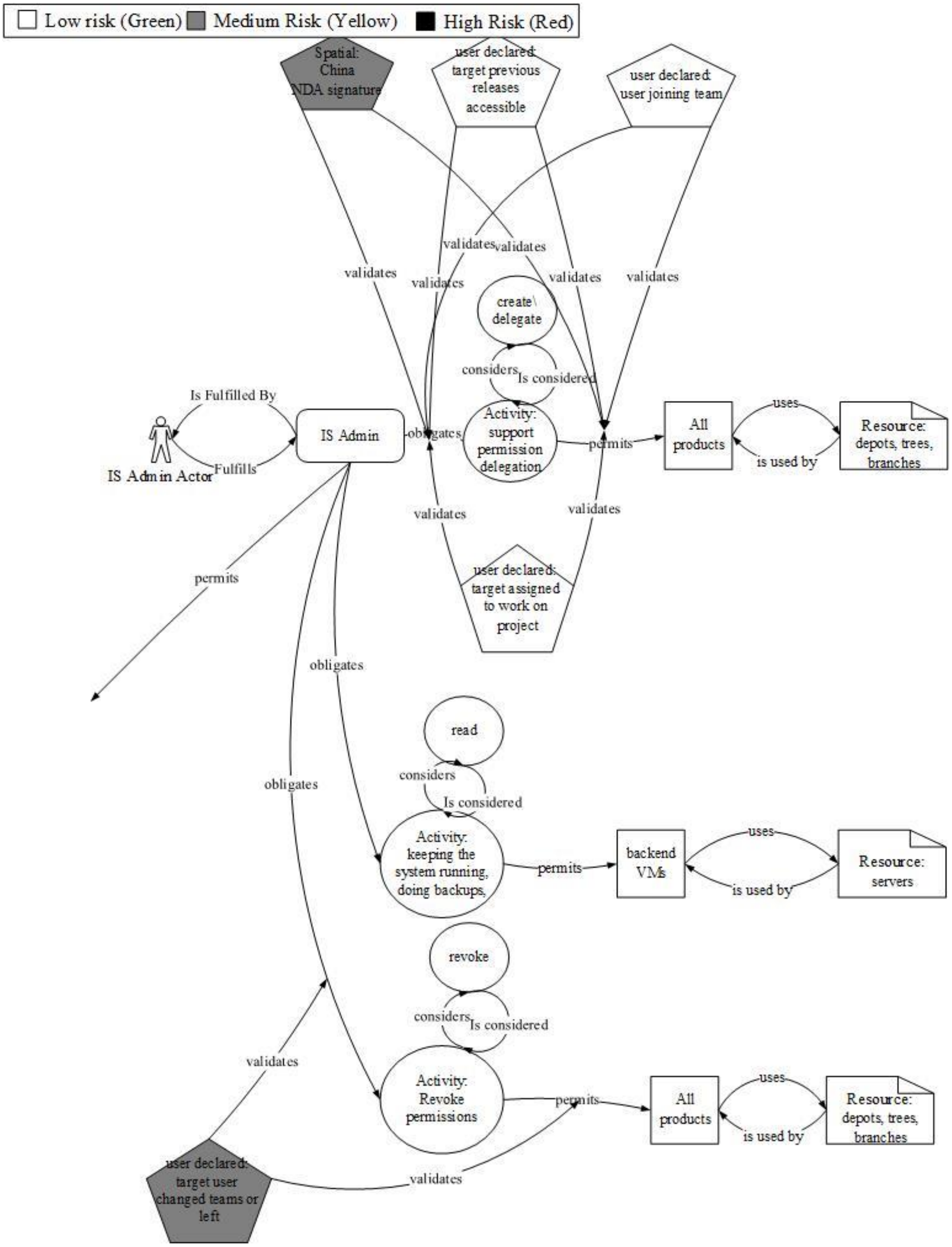

read\ write
Is considered
considers
Is Fulfilled By
Fulfills
Engineering Service Actor
Engineering Service Role
obligates
Activity: Manage servers
permits
Backend hardware
uses
is used by
Resource: Physical servers
permits

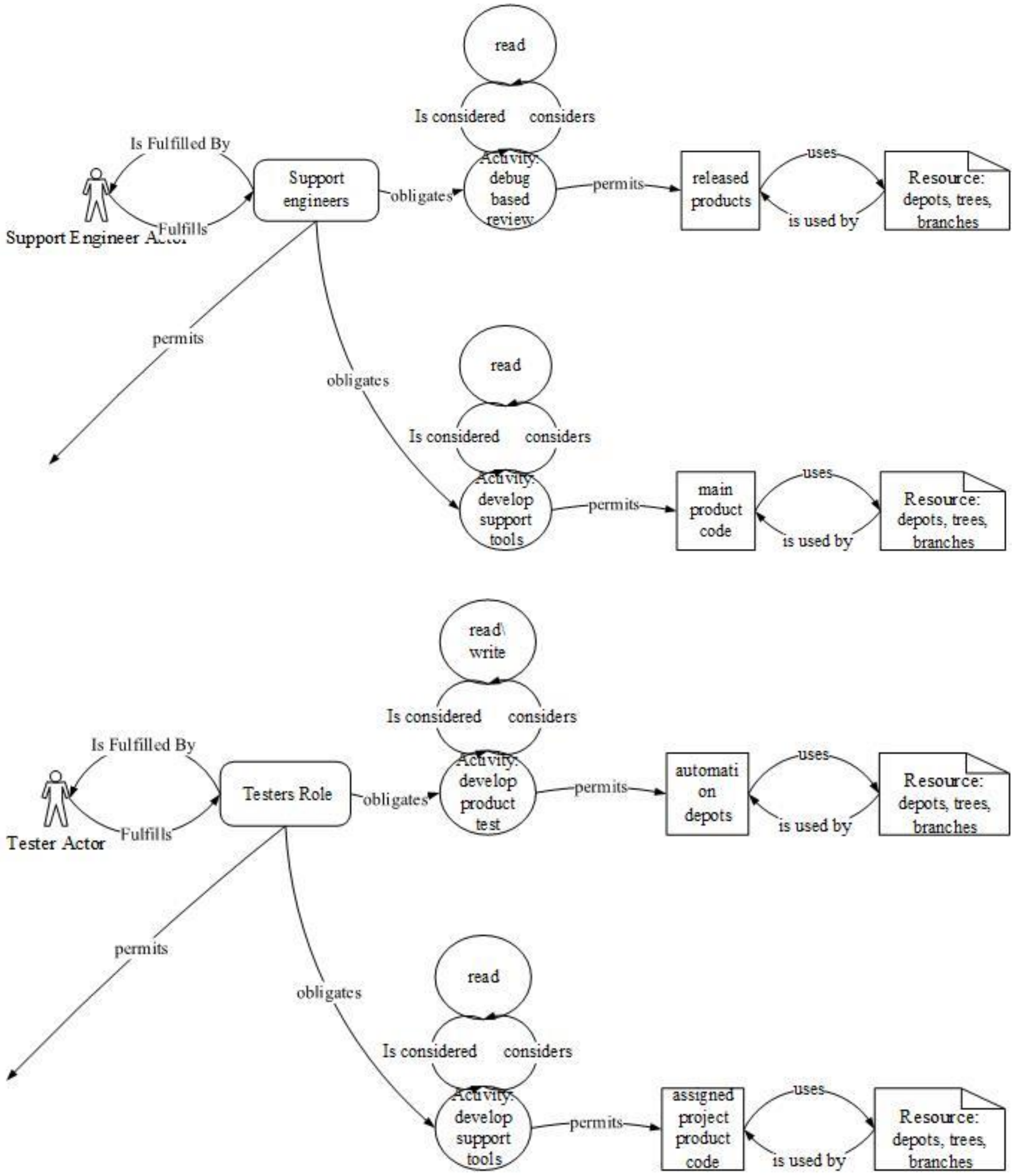

read\ approve
Is considered
considers
Is Fulfilled By
Fulfills
Gatekeeper actor
Gatekeeper Role
obligates
Activity: code review
permits
main product code
uses
is used by
Resource: Code repos
permits

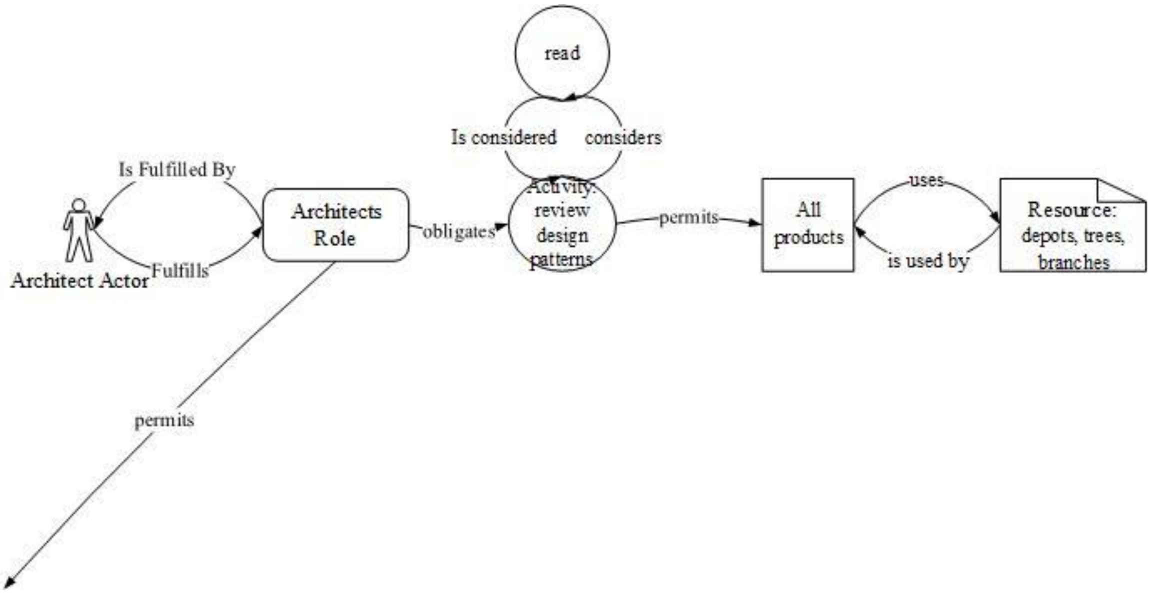

read\ write
Is considered
considers
Is Fulfilled By
Fulfills
Graphics Actor
Grpahics Role
obligates
Activity: manage product graphics
obligates
graphics depots
uses
is used by
Resource: depots, trees, branches
permits

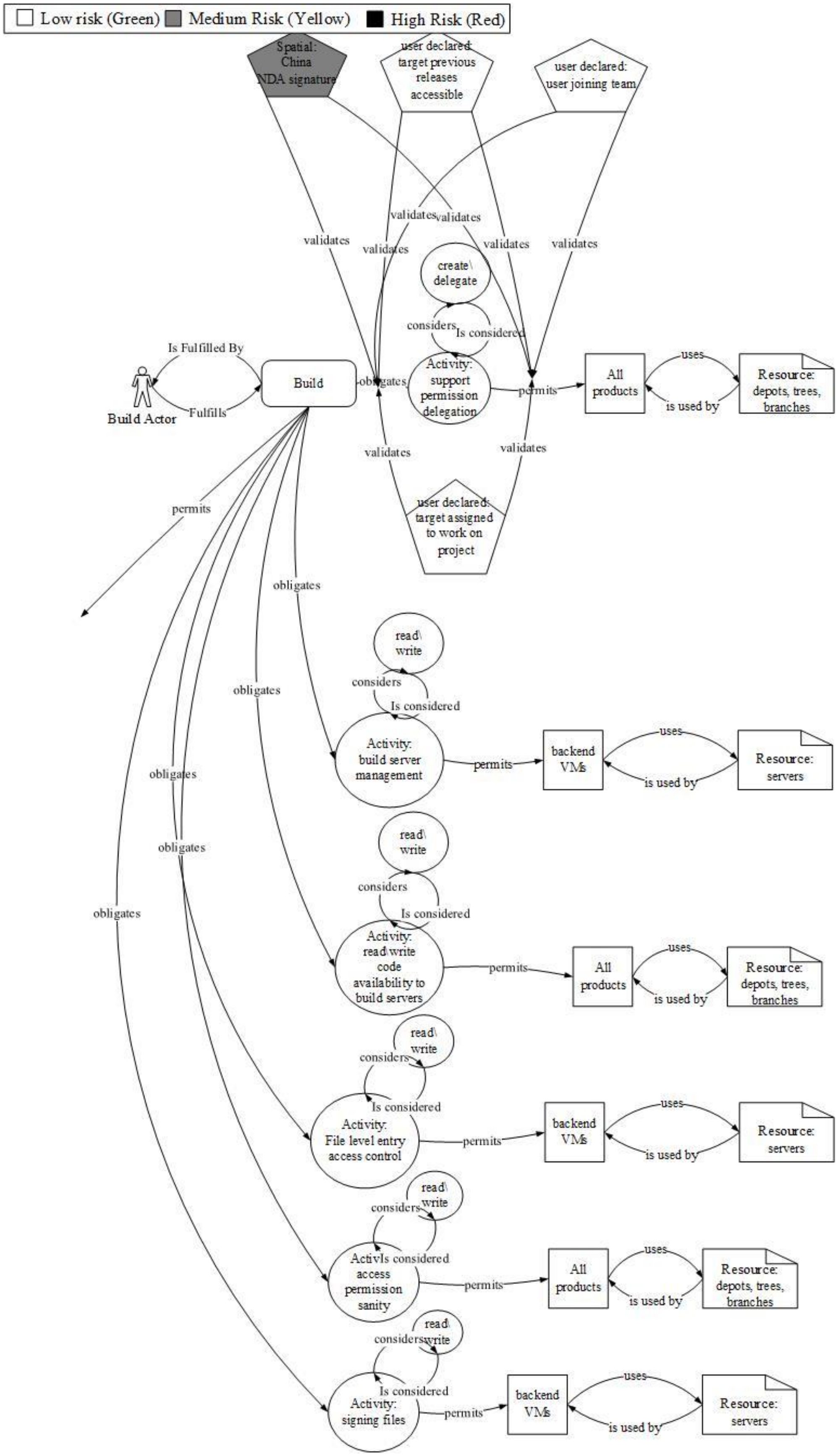

□ Low risk (Green)
▨ Medium Risk (Yellow)
■ High Risk (Red)
Spatial:
User locale
loging
validates
User-declared:
2-way auth
validates
validates
Is considered
considers
Activity:
system
login
n permits
permits
All
Products
uses
is used by
Resource:
depots,trees,bran
ches
validates
validates
validates
User-declared:
account created
validates
Spatial:
remote VPN
validates


□ Low risk (Green) ▨ Medium Risk (Yellow) ■ High Risk (Red)
Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
validates
validates
delegate
considers
Is considered
Is Fulfilled By
Fulfills
Owner\Lead Actor
DevLeads, Managers, Component Owners Role
permits
Activity: team permission delegation
permits
Owned Component
uses
is used by
Resource: depots, trees, branches
user declared: target assigned to work on project
permits
permits
permits
Read\ approve
considers
Is considered
Activity: code review
read\ write
Is considered
considers
Is Fulfilled By
Fulfills
Publications Actor
Publications Role
obligates
Activity: documentation work
obligates
documentation depots
uses
is used by
Resource: depots, trees, branches
permits

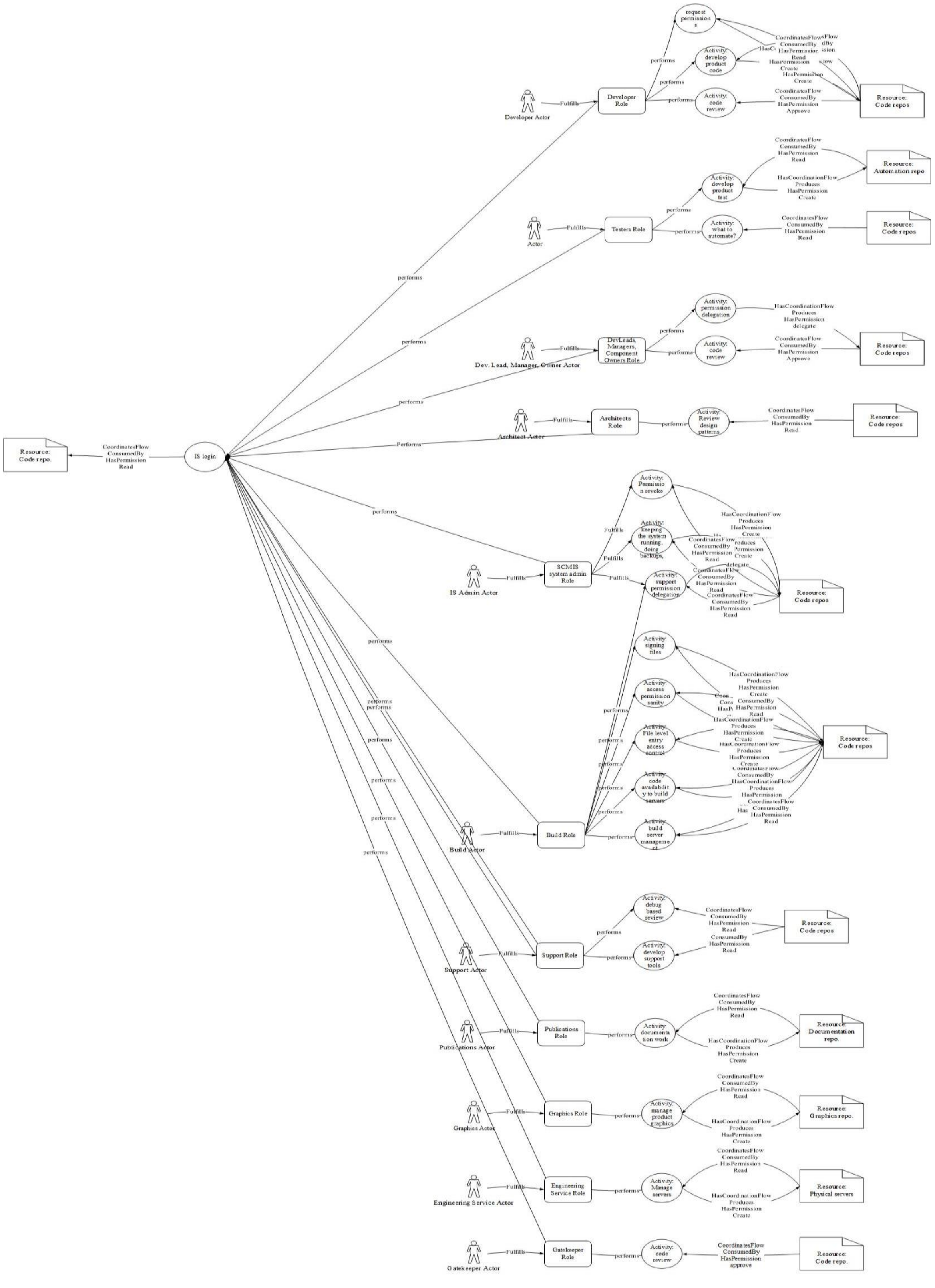

□ Low risk (Green) ▨ Medium Risk (Yellow) ■ High Risk (Red)
Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
create\ delegate
considers
Is considered
Is Fulfilled By
IS Admin
obligates
Activity: support permission delegation
permits
All products
uses
is used by
Resource: depots, trees, branches
IS Admin Actor
Fulfills
validates
validates
user declared: target assigned to work on project
permits
obligates
obligates
read
considers
Is considered
Activity: keeping the system running, doing backups,
permits
backend VMs
uses
is used by
Resource: servers
revoke
considers
Is considered
validates
Activity: Revoke permissions
permits
All products
uses
is used by
Resource: depots, trees, branches
user declared: target user changed teams or left
validates


read
Is considered
considers
Is Fulfilled By
Support engineers
obligates
Activity: debug based review
permits
released products
uses
is used by
Resource: depots, trees, branches
Support Engineer Actor
Fulfills
permits
obligates
read
Is considered
considers
Activity: develop support tools
permits
main product code
uses
is used by
Resource: depots, trees, branches
read\ write
Is considered
considers
Is Fulfilled By
Testers Role
obligates
Activity: develop product test
permits
automati on depots
uses
is used by
Resource: depots, trees, branches
Tester Actor
Fulfills
permits
obligates
read
Is considered
considers
Activity: develop support tools
permits
assigned project product code
uses
is used by
Resource: depots, trees, branches


read
Is considered
considers
Is Fulfilled By
Architects Role
obligates
Activity: review design patterns
permits
All products
uses
is used by
Resource: depots, trees, branches
Architect Actor
Fulfills
permits


□ Low risk (Green) ▨ Medium Risk (Yellow) ■ High Risk (Red)
Spatial: China NDA signature
user declared: target previous releases accessible
user declared: user joining team
validates
validates
validates
validates
validates
validates
create\ delegate
considers
Is considered
Is Fulfilled By
Fulfills
Build Actor
Build
obligates
Activity: support permission delegation
permits
All products
uses
is used by
Resource: depots, trees, branches
validates
validates
user declared: target assigned to work on project
permits
obligates
obligates
obligates
obligates
obligates
read\ write
considers
Is considered
Activity: build server management
permits
backend VMs
uses
is used by
Resource: servers
read\ write
considers
Is considered
Activity: read\write code availability to build servers
permits
All products
uses
is used by
Resource: depots, trees, branches
read\ write
considers
Is considered
Activity: File level entry access control
permits
backend VMs
uses
is used by
Resource: servers
read\ write
considers
ActivIs considered access permission sanity
permits
All products
uses
is used by
Resource: depots, trees, branches
read\ write
considers
Is considered
Activity: signing files
permits
backend VMs
uses
is used by
Resource: servers

SARC Design Artifact


request permissions
Activity: develop product code
Activity: code review
Developer Actor
Fulfills
Developer Role
performs
CoordinatesFlow ConsumedBy HasPermission Read
HasPermission Create
HasPermission Create
CoordinatesFlow ConsumedBy HasPermission Approve
Resource: Code repos
CoordinatesFlow ConsumedBy HasPermission Read
Resource: Automation repo
Activity: develop product test
HasCoordinationFlow Produces HasPermission Create
Actor
Fulfills
Testers Role
performs
Activity: what to automate?
CoordinatesFlow ConsumedBy HasPermission Read
Resource: Code repos
Activity: permission delegation
HasCoordinationFlow Produces HasPermission delegate
Dev. Lead, Manager, Owner Actor
Fulfills
DevLeads, Managers, Component Owners Role
performs
Activity: code review
CoordinatesFlow ConsumedBy HasPermission Approve
Resource: Code repos
Architect Actor
Fulfills
Architects Role
performs
Activity: Review design patterns
CoordinatesFlow ConsumedBy HasPermission Read
Resource: Code repos
Resource: Code repo.
CoordinatesFlow ConsumedBy HasPermission Read
IS login
Performs
performs
Activity: Permission revoke
Activity: keeping the system running, doing backups,
Activity: support permission delegation
IS Admin Actor
Fulfills
SCMIS system admin Role
HasCoordinationFlow Produces HasPermission Create
CoordinatesFlow ConsumedBy HasPermission Read
delegate
CoordinatesFlow ConsumedBy HasPermission Read
CoordinatesFlow ConsumedBy HasPermission Read
Resource: Code repos
Activity: signing files
Activity: access permission sanity
Activity: File level entry access control
Activity: code availability to build servers
Activity: build server management
HasCoordinationFlow Produces HasPermission Create
ConsumedBy HasPermission Read
HasCoordinationFlow Produces HasPermission Create
HasCoordinationFlow Produces HasPermission Create
CoordinatesFlow ConsumedBy
HasCoordinationFlow Produces HasPermission
CoordinatesFlow ConsumedBy HasPermission Read
Resource: Code repos
Build Actor
Fulfills
Build Role
Activity: debug based review
Activity: develop support tools
CoordinatesFlow ConsumedBy HasPermission Read
ConsumedBy HasPermission Read
Resource: Code repos
Support Actor
Fulfills
Support Role
Publications Actor
Fulfills
Publications Role
Activity: documentation work
CoordinatesFlow ConsumedBy HasPermission Read
HasCoordinationFlow Produces HasPermission Create
Resource: Documentation repo.
Graphics Actor
Fulfills
Graphics Role
Activity: manage product graphics
CoordinatesFlow ConsumedBy HasPermission Read
HasCoordinationFlow Produces HasPermission Create
Resource: Graphics repo.
Engineering Service Actor
Fulfills
Engineering Service Role
Activity: Manage servers
CoordinatesFlow ConsumedBy HasPermission Read
HasCoordinationFlow Produces HasPermission Create
Resource: Physical servers
Gatekeeper Actor
Fulfills
Gatekeeper Role
Activity: code review
CoordinatesFlow ConsumedBy HasPermission approve
Resource: Code repo.

References

Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. *Handheld and Ubiquitous Computing*, 304–307.

Aburub, F., Odeh, M., & Beeson, I. (2007). Modeling non-functional requirements of business processes. *Information and Software Technology*, *49*(11), 1162–1171.

Allen, G. N., & March, S. T. (2006). The effects of state-based and event-based data representation on user performance in query formulation tasks. *Mis Quarterly*, 269–290.

Anderson, J. M. (2003). Why we need a new definition of information security. *Computers & Security*, *22*(4), 308–313.

Backhouse, J., & Dhillon, G. (1996). Structures of responsibility and security of information systems. *European Journal of Information Systems, 5(1),* 2–9. Retrieved from http://www.palgrave-journals.com/ejis/index.html

Baskerville, R. L. (1988). *Designing information systems security*. John Wiley & Sons, Inc.

Baskerville, R. L. (1989). Logical controls specification: An approach to information systems security. *Systems Development for Human Progress*, *25*(4), 241–255.

Baskerville, R. L. (1992). The developmental duality of information Systems security. *Journal of Management Systems*, *4*, 1–12.

Baskerville, R. L. (1993). Information Implications Systems Security Design Methods: for Information Systems Development. *ACM Computing Surveys*, *25*(4), 375–414.

Baskerville, R. L. (1994). Research notes: Research directions in information systems security. *International Journal of Information Management*, 385–387.

Baskerville, R. L. (1999). Investigating information systems with action research. *Communications of AIS*, *2*(3), 4.

Baskerville, R. L., & Myers, M. D. (2004). Special issue on Action research in Information Systems: Making IS research relevant to practice. *MIS Quarterly*, *28*(3), 329–335.

Baskerville, R. L., & Wood-Harper, A. T. (1996). A critical perspective on action research as a method for information systems research. *Journal of Information Technology*.

Baskerville, R. L., & Wood-Harper, A. T. (1998). Diversity in information systems action research methods. *European Journal of Information Systems*, *7*(2), 90–107.

Batarseh, F. A., Gonzalez, A. J., & Knauf, R. (2013). Modeling and Using Context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *8175*, 288–301.

Becker, J., Rosemann, M., & Von Uthmann, C. (2000). Guidelines of business process modeling. In *Business Process Management* (pp. 30–49).

Beer, S. (1984). The Viable System Model: Its Provenance, Development, Methodology and Pathology. *Journal of the Operational Research Society*.

Bishop, M. (2003). What is computer security? *Security & Privacy, IEEE*, *1*(1), 67–69.

Blumenthal, S. C. (1969). *Management information systems: a framework for planning and development*. Prentice Hall.

Booysen, H. A. S., & Eloff, J. H. P. (1995). A methodology for the development of secure application systems. In *Proceedings of the IFIP TC11 11th International Conference on Information Security*.

Börjesson, A., Martinsson, F., & Timmerås, M. (2006). Agile improvement practices in software organizations. *European Journal of Information Systems*, *15*(2), 169–182.

Brézillon, P. (2015). Modeling expert knowledge and reasoning in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9405*, 18–31.

Burg, D., Compton, M., Harries, P., Hunt, J., Lobel, M., Loveland, G., … Roath, D. (2014). US cybercrime: Rising risks, reduced readiness Key findings from the 2014 US State of Cybercrime Survey. Retrieved from https://www.pwc.com/us/en/increasing-it-effectiveness/publications/assets/pwc-2014-us-state-of-cybercrime.pdf

Checkland, P. (1981). *Systems thinking, systems practice* (Vol. 1). http://doi.org/10.1016/0143-6228(82)90039-X

Checkland, P., & Holwell, S. (1998). Action research: its nature and validity, *11*(1), 9–21.

Cipriano, C., Zand, A., Houmansadr, A., Kruegel, C., & Vigna, G. (2011). Nexat: A History-Based Approach to Predict Attacker Actions. *Proceedings of the 27th Annual Computer Security Applications Conference on - ACSAC ’11*, 383.

Cuppens, F., & Miège, A. (2003). Modeling contexts in the Or-BAC model. In *Proceedings of 19th Annual Computer Security Applications Conference* (pp. 416–425).

D’Aubeterre, F., Singh, R., & Iyer, L. (2008a). A Semantic Approach to Secure Collaborative Inter- Organizational eBusiness Processes (SSCIOBP). *Journal of the Association for Information Systems*, *9*(3), 231–266.

D’Aubeterre, F., Singh, R., & Iyer, L. (2008b). Secure activity resource coordination: empirical evidence of enhanced security awareness in designing secure business processes. *European Journal of Information Systems*, *17*(5), 528–542.

Dhillon, G. (1997). *Managing information system security*. Macmillan.

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

Dichev, C. (2001). Modeling and Using Context. *Modeling and Using Context*, *2116*, 433–436.

Dobson, J. E. (1990). *A methodology for analyzing human and computer-related issues in secure systems*. University of Newcastle.

Earl, M. J. (1978). Prototype systems for accounting, information, and control. *Accounting, Organizations and Society*, *3*(2), 161–170.

Earl, M. J., Sampler, J. L., & Short, J. E. (1995). Strategies for business process reengineering: evidence from field studies. *Journal of Management Information Systems*, 31–56.

Eikebrokk, T. R., Iden, J., Olsen, D. H., & Opdahl, A. L. (2011). Understanding the determinants of business process modeling in organizations. *Business Process Management Journal*, *17*(4), 639–662.

Elliott, G. (2004). *Global Business Information Technology: An Integrated Systems Approach*. Pearson Addison Wesley.

Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems.

Human Factors: The Journal of the Human Factors and Ergonomics Society, *37*(1), 32–64.

Ferraiolo, D., Cugini, J., & Kuhn, D. R. (1995). Role-based access control (RBAC): Features and motivations. In *Proceedings of 11th annual computer security application conference* (pp. 241–248).

Goldkuhl, G. (2004). Design theories in information systems- a need for multi-grounding. *JITTA: Journal of Information Technology Theory and Application*, *6*(2), 59.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337–356.

Guckenheimer, S. (2016). Our journey to Cloud Cadence, lessons learned at Microsoft Developer Division.

Hammer, M. (1990). Reengineering work: don't automate, obliterate. *Harvard Business Review*, *68*(4), 104–112.

Herrmann, G., & Pernul, G. (1999). Viewing business-process security from different perspectives. *International Journal of Electronic Commerce*, 89–103.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, *19*(2), 4.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design Science in Information Systems Research. *MIS Quarterly*, *28*(1), 75–105.

Hirschheim, R., & Klein, H. K. (1992). Paradigmatic Influences on Information Systems Development. *Advances in Computers*, *34*, 293.

Hirschheim, R., & Klein, H. K. (1994). Realizing emancipatory principles in information systems development: the case for ETHICS. *MIS Quarterly*, 83–109.

Hirschheim, R., Klein, H. K., & Lyytinen, K. (1995). *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations*. *Information Society* (Vol. 13).

Hitchings, J. (1995). Achieving an integrated design: the way forward for information security. In *Information Security-the next decade: Proceedings of 11th International Information Security Conference in South Africa* (pp. 369–383).

Hutchinson, W., & Warren, M. (2000). Using the viable systems model to develop an understanding information system security threats to an organization. In *Proceedings of the 1st Australian Information Security Management Workshop*.

ISOO. (2015). The Information Security Oversight Office. Retrieved from http://www.archives.gov/isoo/

Jaaksi, A. (1998). Jaaksi, Ari. *Journal of Object-Oriented Programming*, *10*, 58–65.

James, H. L. (1996). Managing information systems security: a soft approach. *Proceedings of 1996 Information Systems Conference of New Zealand*.

Järvinen, P. (1997). The new classification of research approaches. *The IFIP Pink Summary-36 Years of IFIP. Edited by H. Zemanek, Laxenburg, IFIP*.

Kalam, A. A. El, Baida, R. El, Balbiani, P., Benferhat, S., Cuppens, F., Deswarte, Y., … Trouessin, G. (2003). Organization Based Access Control. *The Fourth IEEE International Workshop on Policies for Distributed Systems and Networks (Policy)*.

Karyda, M., Kokolakis, S., & Kiountouzis, E. (2001). Redefining information systems security: viable information systems. *Proceedings of the 16th International Conference on Information Security: Trusted Information: The New Decade Challenge*, 453–468.

Klein, H. K., & Hirschheim, R. (1987). Social change and the future of information systems development. In *Critical issues in information systems research* (pp. 275–305).

Krueger, R. A., & Casey, M. A. (2014). *Focus groups: A practical guide for applied research*. Sage publications.

Kubicek, H. (1983). User Participation In System Design: Some Questions About Structure And Content Arising From Recent Research From A Trade Union Perspective. *System Design For, With, and by the Users*, 3–18.

Land, F., & Hirschheim, R. (1983). Participative systems design: Rationale, tools, and techniques. *Journal of Applied Systems Analysis*, *10*(10), 15–18.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*(1), 65–100.

Lindgren, R., Henfridsson, O., & Schultze, U. (2004). Design Principles for Competence Management Systems: A Synthesis of an Action Research Study. *MIS Quarterly*, *28*(3), 435–472.

Loukides, M. (2012). *What is DevOps?* O'Reilly Media.

Malone, T. W. (1987). Modeling coordination in organizations and markets. *Management Science*, *33*(10), 1317–1332.

March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251–266.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly*, 179–212.

Mårtensson, P., & Lee, A. S. (2004). Dialogical action research at Omega Corporation. *MIS Quarterly*, *28*(3), 507–536.

McDermott, J., & Fox, C. (1999). Using abuse case models for security requirements analysis. In *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual* (pp. 55–64).

Mumford, E. (1981). Participative Systems Design: Structure and Method. *SYS. OBJECTIVES, SOLUTIONS.*, *1*(1), 5–19.

Myers, M. D. (1997). Understanding Context Before Using It. *MIS Quarterly*, *21*, 241.

Naumann, J. D., & Jenkins, A. M. (1982). Prototyping: the new paradigm for systems development. *Mis Quarterly*, 29–44.

Newell, A., & Simon, H. A. (1972). *Human problem solving* (Vol. 104).

Oh, S., & Park, S. (2003). Task--role-based access control model. *Information Systems*, *28*(6), 533–562.

Olnes, J. (1994). Development of security policies. *Computers & Security*, *13*(8), 628–636.

Ould, M. A., & Ould, M. A. (1995). *Business Processes: Modeling and analysis for re-engineering and improvement* (Vol. 598). Wiley Chichester.

Peffers, K., Tuunanen, T., Rothenberger, M. a., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45–77.

Pernul, G. (1992). Security constraint processing during multilevel secure database design. In *Computer Security Applications Conference, 1992. Proceedings., Eighth Annual* (pp. 75–84).

Peters, S. (2009). 14th Annual CSI Computer Crime and Security Survey. Retrieved from http://www.acis.pamplin.vt.edu/faculty/wallace/5584/CSISurvey2009.pdf

Porter, M. E., & Millar, V. E. (1985). How information gives you competitive advantage. Harvard Business Review, Reprint Service.

Pries-Heje, J., Baskerville, R., & Venable, J. (2008). Strategies for design science research evaluation. *ECIS 2008 Proceedings*, 1–12.

Puhakainen, P., & Siponen, M. (2010). Improving employees' compliance through information systems security training: an action research study. *Management Information Systems*, *34*(4), 757–778.

Raghu, T. S., & Vinze, A. (2007). A business process context for Knowledge Management. *Decision Support Systems*, *43*(3), 1062–1079.

Ricci, A., Omicini, A., & Denti, E. (2002). Virtual enterprises and workflow management as agent coordination issues. *International Journal of Cooperative Information Systems*, *11*(03n04), 355–379.

Röhm, A. W., Pernul, G., & Herrmann, G. (1998). Modeling secure and fair electronic commerce. In *Computer Security Applications Conference, 1998. Proceedings. 14th Annual* (pp. 155–164).

Rouge, C. M. Le, & Niederman, F. (2006). Information Systems and Health Care XI: Public Health Knowledge Management Architecture Design: A Case Study. *Communications of the Association for Information Systems*, *18*(1), 2–54.

Rubin, H. J., & Rubin, I. S. (2011). *Qualitative interviewing: The art of hearing data*. Sage.

Sedlack, D. J. (2012). *Reducing Incongruity of Perceptions Related to Information Risk : Dialogical Action Research in Organizations by*. Nova Southeastern University.

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *MIS Quarterly*, 37–56.

Simon, H. A. (1996). *The sciences of the artificial* (Vol. 136). MIT press.

Singh, R., & Salam, A. F. (2006). Semantic information assurance for secure distributed knowledge management: A business process perspective. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *36*(3), 472–486.

Singh, R., Salam, A. F., & Iyer, L. (2005). Agents in e-supply chains. *Communications of the ACM*, *48*(6), 108–115.

Siponen, M. (2002). *Designing Secure Information Systems and Software (Doctoral Dissertation), University of Oulu*. University of Oulu.

Siponen, M., Baskerville, R., & Heikka, J. (2006). A Design Theory for Secure Information Systems Design Methods. *Journal of the Association for Information Systems*, *7*(11), 725–770.

Stewart, D. W., & Shamdasani, P. N. (2014). *Focus groups: Theory and practice* (Vol. 20). book, Sage Publications.

Straub, D. W., & Welke, R. J. (1998). Coping with systems risk: security planning models for management decision making. *MIS Quarterly*, 441–469.

Strauss, A., & Corbin, J. (1998). *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Sage Publications.

Stringer, E. (2013). *Action Research*. SAGE Publications.

Susman, G. I. (1983). Action research: a sociotechnical systems perspective. *Beyond Method: Strategies for Social Research*, 95–113.

Tan, W., Shen, W., Xu, L., Zhou, B., & Li, L. (2008). A business process intelligence system for enterprise process performance management. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *38*(6), 745–756.

Tejay, G. (2008). *Shaping Strategic Information Systems Security Initiative In Organizations (Doctoral Dissertation)*. Virginia Commonwealth University.

Tremblay, M. C., Hevner, A. R., & Berndt, D. J. (2010). Focus groups for artifact refinement and evaluation in design research. *Communications of the Association for Information Systems*, *26*, 1.

Vaishnavi, V., & Kuechler, B. (2004). Design Science Research in Information Systems.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, *3*(1), 36–59.