DEMO and Security

An investigation of the connections between Security and the Design & Engineering Methodology for Organizations (DEMO)

http://www.demo.nl

Ioana Marginas

TU Delft
Delft University of Technology

IBM

DEMO and Security

An investigation of the connections between Security and the Design & Engineering Methodology for Organizations (DEMO)

www.demo.nl

Master's Thesis Project
Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Information Architecture Group
August 2006

Thesis' Committee	**Prof. Dr. Ir. J.L.G. (Jan) Dietz (TU Delft)** **Ing. R.G. (Rob) Weemhoff (IBM, The Netherlands)** **Prof. Dr. R. W. (René) Wagenaar (TU Delft)** **Prof. Dr. A. (Arie) de Bruin (TU Delft)**
Author	**I.L. (Ioana) Marginas, Licentiate**
Mentors	**Prof. Dr. Ir. J.L.G. (Jan) Dietz** **Ing. R.G. (Rob) Weemhoff**
Research sponsor	**IBM Netherlands N.V.**
Period	**December 2005 – August 2006**

Abstract

This thesis is aimed at exploring how security aspects within organizations can be addressed at a very high level: an ontological level that encapsulates construction and operation issues of organizations with no reference to implementation concerns. To do this, DEMO (Dynamic Engineering and Modeling for Organizations) has been found as the relevant methodology to use.

The thesis has mainly four contributions. (1) First, it identifies the thread that connects DEMO with security. It does that by performing a thorough study of information systems security issues and DEMO. The research brings forward the current state in the information systems security field and concludes by pointing out the connection between DEMO and security - responsibility. (2) Second, based on the results of the previous investigation, it analyses various approaches to model security starting from responsibility with emphasis on their strengths, week points, similarities and differences. (3) Third, it performs a critical analysis of DEMO from a security perspective. The findings are analyzed and discussed and DEMO's approach to responsibility is compared with the previous analyzed security modeling approaches based on responsibility. The results of this comparison constitute the (4) fourth contribution of the thesis: a starting point for modeling security within DEMO. Two case studies will be used for illustration purposes of the proposed method.

To Marius

Acknowledgements

I would like to express my gratitude to everyone who helped me complete my thesis work. First and foremost, my thanks go to my university supervisor, Professor Dr. Ir. Jan L.G. Dietz, and to my IBM mentor, Ing. Rob G. Weemhoff, for all their constant help and guidance.

My thanks also go to Mr. Martin Op 't Land whose practical inputs and discussions proved really useful along the way. I would also like to thank Professor Dobson for the efficient and valuable exchange of ideas and e-mails. From TUDelft, I would like to offer my gratitude to Ir. Frans Ververs for all the help and support throughout the whole master's programme. From IBM's Center for Advanced Studies, I would like to thank Jasper Schroder who was always willing to provide an answer to any kinds of questions I had. I also wish to thank my fellow students from IBM's Center for Advanced Studies in Amsterdam for all the pleasant lunches and coffee breaks we had. My appreciation would be incomplete without thanking my family and friends for all their support during the last two years.

Ioana Marginas
Amsterdam, August, 2006

.

Table of contents

Defining the research questions is probably
the most important step taken into a research study [Yin94].

1 Introduction

The enterprise of today is mainly characterized by complexity. This complexity triggers the need of a thorough understanding of its organization, construction and operation when one attempts to model it. But today's most used methodologies look at the organization from a much too technical-oriented perspective or a much too organizational-oriented one. Due to the former, architectural crosscutting concerns such as security end up to be mainly technology driven, being attained through a bottom-up approach rather than being identified starting at a very high level and the technology to be just the one to solve them. Due to the latter, the accent is mainly set on governmental issues and it is hard to extract the core that would drive one to constructively model concepts such as security.

DEMO [demo] methodology for (re)designing and (re)engineering organizations is based on the Ψ[1]-paradigm (read PSI). Conform to this theory, "the operational principle of organizations is the ability of human beings to enter into and comply with commitments towards each other, collectively called social interaction". This way, the theory creates the main difference between organizations and other kinds of systems (i.e. information systems) [demo]. Thus, in a DEMO perspective, "an organization is a social system of successful and interrelated business transactions" [Die99a]. Representing the enterprise using DEMO implies creating a conceptual model of the enterprise, also called ontology[2], which is completely independent from all realization and implementation issues [Die05a]. The model has the major advantage of representing the essence of the operation of an enterprise. It does this by complying with five major properties: it is *coherent* (it is a logical and integral whole), *comprehensive* (all the aspects are covered, nothing is left out), *consistent* (there are no irregularities or contradictions), *concise* (they live out the fast changing aspects of an organization like the technological ones) and, the most important of all, is *essential* (they do not show any unnecessary detail, concentrating on the core of the enterprise, on its essence, its deep structure). Through these five properties, DEMO becomes a high level modeling approach making the aspects of an organization very easy to communicate to managers that "do not want to be confused with unnecessary details".

If one thinks of security from a business point of view and not in terms of firewalls and cryptology algorithms, one might find DEMO as being suitable for attaining a high understanding of security aspects, for making security aspects understandable at a management level. The characteristics of DEMO and a thorough knowledge of security drove Mr. Rob Weemhoff from IBM to the idea of trying to model security using DEMO as modeling technique. This, and the fact that DEMO has as main promoter Professor Jan Dietz from Delft University of Technology, constitute the premises of a master level research project within TU Delft and IBM with the theme *DEMO and Security*.

The remainder of this Chapter will introduce the main research goal of such a research, will define the research questions to be answered and establish the research methodology most suitable for such a research. The Chapter will end by providing the reader with an outline of this thesis.

1.1 Research background

Traditional approaches to security are mainly based on the so called "penetrate and patch" technique: whenever a vulnerability of a system was noticed, the weaknesses were identified and removed [Jur05]. It is now known that these kinds of approaches were proven to be not appropriate as by the time the vulnerability was detected much harm had already been done.

[1] Performance in Social Interaction
[2] The notion of enterprise ontology will be presented and discussed in Chapter 2.3

Therefore, the need of considering security in the early stages of the system life cycle was identified. Various attempts designing security methods with the help of mathematical mechanisms and tools are referenced in [Jur05]. All of them have the final goal of establishing "crucial requirements at the specification level through formalization and proof, which may be mechanically assisted or even automated". Although a model for security purposes is proven to be useful, problems were detected in the sense some specifications were detected just by trying to make them "sufficiently precise for formal analysis".

The need of having security as concern during each step of designing a system was identified [Dev00] and therefore top-down approaches to security have been attempted. To this end, we refer to [Lie97] as a useful approach in documenting information security requirements by organizing them in a top-down layered manner.

The above presented approaches are very much based on technical aspects. They all attempt to look at security aspects from the technical perspective. We include UML and its UMLsec extension in this category. But none of them has managed to surpass the implementation line. DEMO looks at a system from the point of view of its construction and operation completely independent of its implementation. That is what makes DEMO different and here resides the motivation of studying security at a DEMO level: a study of security at a level that encapsulates construction and operation issues with no reference to implementation concerns.

1.2 Research questions

As concluded from the research background, there is a need of studying security issues at a DEMO level. Therefore, the main research goal of the thesis is to investigate the possibility of addressing security concepts within an organization seen as "a social system of successful and interrelated business transactions" [demo]. The final purposes would be to identify the thread that makes the connection between DEMO and security, to analyze DEMO from a security perspective and to establish a starting point for addressing security with the help of DEMO's way of modeling organizations. To achieve the goal, the following research questions are proposed:

First, in order to establish the path that will help us to identify the possible connection between Security and DEMO, the following question needs to be answered:

1. To what extent does the current research in the field of security address and solve the information systems security concerns?

With the results of the first question and the knowledge of DEMO, the second question needs to be answered next:

2. What is the connection thread between DEMO and Security?

The third question comes as a consequence of the second one. As we will see, the connection thread between DEMO and Security is identified as the concept of 'responsibility'. Therefore, the third question comes as the next natural question to be asked:

3. To what extent does the current research address the issue of 'responsibility – a security issue'?

Question number 4 has to do with the need of having a critical analysis of DEMO from a Security driven perspective in order to see how security can be addressed within DEMO:

4. Which are the security aspects incorporated in the DEMO methodology?

Establishing a way of addressing security within DEMO makes the purpose of the last question:

5. How could security aspects be tackled within the DEMO methodology?

1.3 Research strategy and approach

In order to answer the research questions and to attain the research goal of this thesis, it is important to carry out the research activities following systematic scientific research methodologies. To this end, [Yin94] illustrates several research strategies, like experiments, surveys, archival analysis, history and case study. Each strategy can be used for three purposes (exploratory, descriptive, or explanatory) and each is presented with its logic, its way of collecting facts and data, and with its advantages and disadvantages.

According to the theory presented in [Yin94], there are three conditions that govern the selection of a proper research strategy; these are: the type of research questions; the control the researcher has over actual behavioural events; the degree of focus on contemporary as opposed to historical events.

Taking into consideration the research strategies presented and discussed in [Yin94] and also in [Gal92], and given the characteristics of this thesis, the following combination of research strategies is proposed: literature study, survey research (discussions with experts) and case study. For a more detailed view on research strategies, we refer the reader to [Yin94] and [Gal92].

Following the previous discussion on the importance of a sound research strategy when carrying out a research project, it is also important to have a well defined, step-by-step approach of how the research will be conducted. Therefore, the following phased strategy is proposed:

Phase 1:
Literature study: topics like information systems security, security and ontology, enterprise ontology, and DEMO are examined. This phase ends with establishing the general direction of the research: it identifies the connection thread that ties DEMO and Security and it establishes the next steps to be taken in the following phases.

Phase 2:
Analyze the theoretical foundation necessary for the research: security approaches based on responsibility. In this phase, the main concern is the study and analysis of the existing responsibility approaches to security and also of those methods that base their approach on high level management of security issues within an enterprise. The identified methods are analyzed, discussed and compared.

Phase 3:
Critical analysis of security issues incorporated by DEMO: identify, analyze and discuss the security elements incorporated by DEMO. This phase also deals with a comparison between the way DEMO encloses and deals with various aspects of security and the other mentioned approaches (the ones identified in phase 2). Next to this, it analyses how the experience gained from the existing research of 'responsibility – a security issue' can be used to attain a way of modeling security within the DEMO context.

Phase 4:
Apply and analyze the theory on study cases: the Library case is taken as illustrative example and the Import/Export Service Company as a real life example.

Phase 5:
Write the thesis: this phase is mainly dedicated to write the results obtained in all the previous phases and provide a report of the work performed.

1.4 The structure of the thesis

Following the research strategy proposed in the previous Section, the outline of this thesis is as follows. In Chapter 2, the theoretical background needed to support the research is offered. Topics as a survey of the current information systems security methods, the relationship security-ontology, the concept of enterprise ontology and the DEMO methodology constitute the subjects of this Chapter. This way, the first two questions of the thesis will be answered. Current responsibility modeling approaches to security are presented in Chapter 3; here, the third research question is answered. Chapter 4 comes with a critical analysis of the security issues incorporated in DEMO, providing the answer for the forth question. The proposed solution to the research objective is offered at the end of this Chapter (question 5), while Chapters 5 and 6 analyze the theory on two case studies, namely the classical library example portrayed within the DEMO methodology and the case of an Import/Export Service Company. Discussions of the findings, limitations of this thesis and implications for potential future research are presented in Chapter 7. The report draws to an end in Chapter 8 where the work is summarized and concluded.

A wise man hears one word and understands two (Yiddish Proverb):
Responsibility - the thread that connects DEMO and Security

2 Theoretical background for the work

This Chapter introduces the core concepts necessary for conducting this research. First, to identify the possible approaches that could be used when addressing security at DEMO level, a survey of the current information systems security methods has been performed. The first parts of the Chapter bring forward the results of the research performed and, this way, provide the answer to the first research question: to what extent does the current research in the field of security address and solve the information systems concerns. Section 2.2 introduces the relationship between ontology and information security. In the last two Sections, the concept of enterprise ontology and the DEMO methodology are explained as they are described in [Die05a], just enough to make the reader understand the rest of the report. The Chapter ends with a discussion over the findings.

Although research in information systems security
has moved towards a socio-organizational perspective,
the approaches for managing security still reside on the idea of
the "organization as a machine" [Dhi01b].

2.1 A survey of information systems security approaches

There are several studies that tackle the classification and comparison of information security methods based on generations. The author of [Sip05a] attempts a first classification of traditional security methods, as follows: checklists, information systems security standards, information systems security maturity criteria, risk management, formal methods. The author considers that although included in the category of traditional approaches, the methods he lists are commonly used in practice and they get much attention from scholars and from practitioners.

General information system design methods are divided into three generations that are considered to constitute a framework for "comparing and understanding current security design methods" [Bas93]. This way, back in 1993, the authors established three generations of information security methods; these are:

1) First generation

The first generation (1972-) is the one steered by checklists methods (i.e. security checklists and risk analysis). These are methods that "map limited solutions onto the information problem". The main features are described in Table 2-1, from [Bas93]. For a detailed explanation we refer the reader to [Bas93].

First generation – Checklist Methods			
Assumptions	**Activities**	**Strengths**	**Weaknesses**
• The boundaries of the solution space is most clearly defined by a highly limited useful solution • Each member of that set of solutions will be universal to a large degree • The ultimate benefits of a system control can be expressed by a lower probability of a threat occurrence or by the mitigation of the expenses caused by such an occurrence	• Survey checklist of the security items • Risk analysis • Implementation	• Thorough examination of possible security components • Lower expertise and training • Low cost • A wide variety of computer-based support tools	• Oversimplify more complex information systems • Proclivity for unauthorized design shortcuts • Documents are hard to understand and difficult to maintain • High maintenance costs • Overlook any innovative or new solutions heavy dependence on risk quantification

Table 2-1: First generation of information security approaches [Bas93]

2) Second generation

The second generation (1981-) belongs to the mechanistic engineering methods (i.e. CRAMM, BDSS, control point and exposure analysis matrices, computer questionnaires). These are methods that use a "partitioned complex solution that matches with the functional requirements." The main features are described in Table 2-2, from [Bas93]. For a detailed explanation we refer the reader to [Bas93].

Second generation – Mechanistic Engineering Methods			
Assumptions	**Activities**	**Strengths**	**Weaknesses**
• The requirements and impacts of security systems elements will be complete and interconnected • The exact controls could be unique and ideals • A feasible solution set in not bounded • A well understood well-documented security design leads to efficient security maintenance and modification	• Inventory assets and threats • Enumerate possible controls • Risk analysis • Prioritize controls • Implement and routine review	• Comprehensiveness of the approach • Detailed and well-organized documentation • Reduced operation and maintenance costs • Useful for every complex systems • Impact of modifications easily assessed	• Complex design process • High degree of training required • A design team needed • Higher costs • Preexisting system or specifications • The functional and security design are isolated • Irrational exposure estimates

Table 2-2: Second generation of information security approaches [Bas93]

3) Third generation

The third generation (1988-) is guided by the Logical-Transformational methods (i.e. logical controls design, data flow diagrams). These are methods that use a "highly abstracted design for expressing the problem solution and space". The main features are described in Table 2-3, from [Bas93]. For a detailed explanation we refer the reader to [Bas93].

Third generation – Logical-Transformational Methods			
Assumptions	**Activities**	**Strengths**	**Weaknesses**
• Ideal security solutions will evolve only from an understanding of the broad problem situation • Abstract models clarify the organizational problems, and more effective controls will result • Design founded on such problems will prove flexible, adaptable and consequently linger-lived • There are few universal solutions • Controls place constraints on information systems	• Model building • Stakeholder analysis • Translation of abstract models into reality • Implementing physical models • Maintaining	• Flexibility and controls • Excellent documentation • Low maintenance costs • Closes connection • Lessen the relevance of risk analysis • A concerted security functionality design process • Less conflict between security and system	• Lack of experience • Abstract controls are difficult • Physical security details are postponed • Difficult cost evaluation • Not as useful for existing systems

Table 2-3: Third generation of information security approaches [Bas93]

The analysis preformed in [Bas93] revealed the idea of a possible juncture between security methods and more general information systems methods. But at the same time, a problem is identified: when security is modeled using the more general methods' framework, the security specifications fail to be rigorously considered. To this end, [Bas93] identifies the need for security methods to be integrated in more general methods; this way, "express means for analyzing and

designing security controls" are provided with the final scope of gaining the practical usage of these more general security methods.

Twelve years after the above generations' analysis, [Sip05b] comes with another information systems security generations' division. This generations' split is partially portrayed by Figure 2-1 (partially, because the fifth generation is not represented). To understand the reasons behind this division, we will first describe the way of working in [Sip05b]. The generations' description comes next.

In performing his work, the author of [Sip05b] uses an analysis framework based on four viewpoints; these are: research objectives (critical, interpretative and means-end oriented), organizational role of information systems security (technical, socio-technical and social), research approaches and methods used (mathematical, conceptual-analytical, empirical theory testing, empirical theory crating, artifact building or evaluating) and applicability to information system development (applicable to certain information system development method, potentially applicable, not applicable).

The second viewpoint is considered to be the most relevant for our research as the organizational role of information systems security (ISS) is about the relationship between information systems security and the organization [Sip05b]. To this end, Table 2-4, from [Sip05b], portrays the three possible views of information systems security (technical, socio-technical and social) with regard to four information systems security concerns (ISS design priority, cause of ISS problems, ISS design success, and users' impact on ISS design). In this approach (Table 2-4), one can observe that in the technical view, security is regarded only from the technical point of view whereas the social view points out that the organizational systems have to be developed before technical issues. The latter means that the "key to ISS design success is social-organizational desirability and meeting users' preferences" [Sip05b]. The socio-technical view bridges the other two by coming with compromises when necessary (i.e. the security requirements conflict with the users' preferences). The author brings together various literature examples in his attempt to say that security problems may rise from the "lack of fit" between social and technical approaches: security "is not just a technology problem, but it also involves people". His reasoning is aimed towards the idea of a next generation of "social and adaptable information systems security methods".

Four information systems security (ISS) design concerns	**Technical view**	**Socio-technical view**	**Social view**
ISS design priority	Technical system	Technical and social-organizational systems are equally important	Social-organizational systems
Cause of ISS problems	Technical quality and user resistance	Poor fit between the technical and social-organizational systems	Social disinterest
ISS design success depends	Technical quality	Good fit between technical and social-organizational system	Socio-organizational desirability
Users' impact on ISS design	Users have no or very little impact on ISS design; user control	Finding a fit or consensus between users' preferences and technical security concerns. Users have moderate impact on ISS design.	Users' preferences are the first priority: users have huge impact on ISS design

Table 2-4: The organizational roles of information systems security [Sip05b]

Having explained the way of working in [Sip05b], the reader is now directed to a brief generations' description (see Figure 2-1).

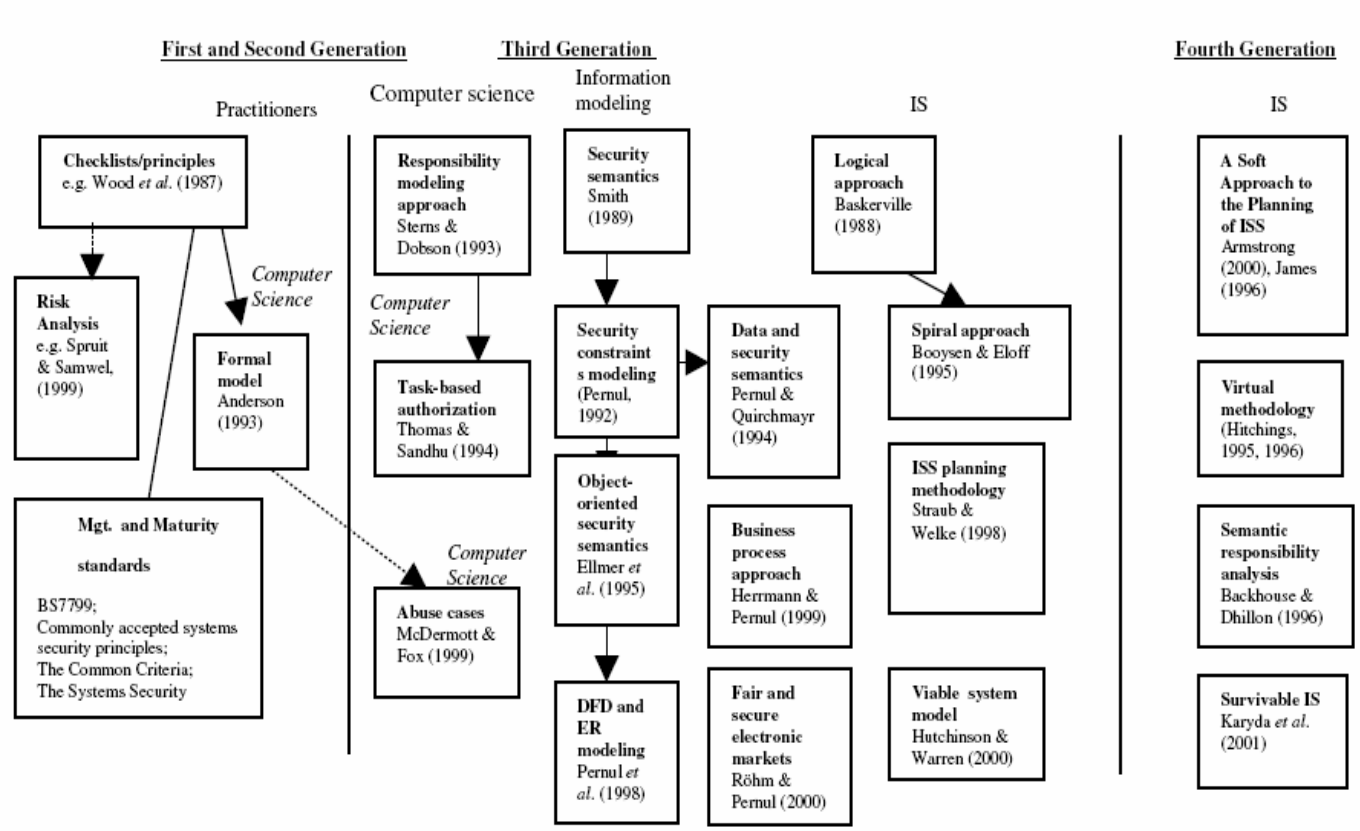
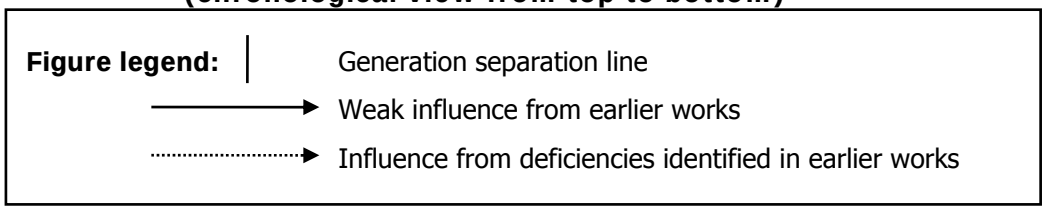


Figure 2-1: Information systems security generations [Sip05b] (chronological view from top to bottom)



The first two generations' methods are the traditional ones and they have been listed in Section 2.1. An in depth analysis of these methods is carried out in [Sip05a]. The conclusion of the work performed is that in traditional information systems security approaches (first and second generations), the methods "entail the technical view of the organizational role of information systems security" whereas the social-organizational nature is not seriously considered. To this end, the author identifies the need of social information systems security methods.

In [Sip05b], the author introduces 3 more generations of information systems security methods; these are:

- the third generation: includes methods for modeling organizations' ISS requirements (i.e. logical modeling, spiral approach, planning methodologies, the responsibility modeling approach by Dobson, the task-based authorization, the abuse case etc). All of these methods are thoroughly enumerated and referenced in [Sip05b];
- the fourth generation: includes methods that add the socio-technical design aspect to the ones in the third generation (i.e. the user participation utilized in the soft approach to the planning of ISS, semantic responsibility analysis, survivable IS approach); all of these methods are enumerated and referenced in [Sip05b];

In the same work, M. Sipponen claims that the current ISS development is in its fourth generation and that the future methods will combine into a fifth one:

- the fifth generation: includes methods that should "encompass social and adaptable ISS methods that are rigorously developed along with practice". Hence, these methods should integrate social techniques (i.e. user's participation) in order to guarantee the social acceptance of technical procedures and they should also be integrated with different information systems development methods. To prove their applicability they also should be tested in practice; this means that their development will need practitioners' input.

The work in [Sip05b] concludes with the statement that all current ISS methods (first four generation) should move towards a fifth generation: "the social and adaptable ISS methods".

The work in [Dhi01b] tries another approach than in [Sip05b] by analyzing the socio-philosophical concerns from different information systems and security approaches. The authors argue that in order for security approaches to work in a "systematic and appropriate manner", it is essential to understand their "conceptual basis". To this end, in [Dhi01b] the research is based on a conceptual framework that includes four paradigms:

- the functionalist paradigm: incorporates an objective view "concerned with the regulations and control of all organizational affairs"; "practical solutions to practical problems"; theories used: system theory and contingency theory; security methods: traditional risk analysis approaches, security evaluation methods;
- the interpretive paradigm: studies "the world as it is"; the social reality is "a network of assumptions and inter-subjectively shared meanings" and reality is a result of individuals' actions; theories used: structuration theory, phenomenology and Hermeneutics, semiotics, conceptualism; security methods: risk analysis and communicative content, speech act theory and security development, pragmatic considerations and security;
- the radical humanist paradigm: views the society as antihuman stressing the emancipation of human "beings so that they may realize their full potential"; theories used: critical theory, anarchistic individualism; security methods: strategic options for security, critical theoretic considerations in risk analysis;
- the radical structuralist paradigm: advocates radical change while sharing the functionalists' objective view point; the key notion is that "change in society inevitably involves a transformation of structures which, even given favorable circumstances, do not fall or change in their own accord"; theories used: conflict theory; methods used: not found;

Without going into more depth regarding the above paradigms, the reader is referred to [Bur79]. In what follows, we want to bring forward the security related contributions within the interpretive paradigm as we consider them to be the most closed to our interests. To this end, [Dhi01b] performs a good analysis of the security approaches in the interpretive paradigm. The authors claim that although research in information systems security approaches has moved towards a socio-organizational perspective, the approaches for managing security still reside on the idea of the "**organization as a machine**" and they do not consider users' interests. Research directions within this paradigm include works that are based on the speech act theory (as portrayed in [Sea69]) to specify organizational security requirements (i.e. [Dob91], [Str93]). As DEMO also incorporates the speech act theory we consider this approach as subject for further study. Another work is the one in [Bac96] where security is looked at as "an outcome of communication breakdowns". In doing so, [Bac96] uses semiotics to "interpret the security implications of organizational actions". Both works base their approach on analyzing responsibilities within an enterprise and starting from this analysis they are trying to deal with security concerns within an organization.

The research carried out in [Dhi01b] draws to the overall suggestion that the socio-organizational perspective constitutes the future way for approaching security of information systems. As this perspective is incorporated within the interpretive paradigm which also includes works within the speech act theory (as does DEMO), we consider the research in this direction as

a staring point for our work. Therefore, security approaches based on responsibility analysis will be further considered for analysis and discussion.

2.2 Ontology and Information Security

As also argued in the previous paragraph, most attempts in dealing with information security find their routes in the technical field. As the technology advances - and it always did and will always do - new systems are developed, new vulnerabilities appear, new threats are posed and new ways of dealing with them are needed. Therefore, the need of a broader, more general context in which to explain and talk about information security has emerged.

To this end, it has been argued by both scholars and practitioners that what the information security field needs is an ontology: a collection of the most important security concepts with their interpretations and the relationships between them [Don03]. Adopting a more practical definition, the authors in [Ras2002] refer to ontology in this context as a "highly structured system of concepts covering the processes, objects and attributes of a domain" and all their relations. Their work introduces an ontological semantic approach to information security, considered by the field's community as a "powerful means to organize and unify the terminology and nomenclature".

The literature mentions the Orange Book [ORANGE], the US Department of Defense Standard, as a relevant, widely used and generally accepted security book. Its purpose was to provide "technical hardware/firmware/software security criteria and associated technical evaluation methodologies in support of the overall system security policy". Although a good starting point, the Orange Book proved to be not very useful when the problems of computer networking appeared. Thus, the technological changes have shown their power again.

Since then there have been other attempts to construct pertinent glossaries. The experts' attempts in unifying and finding a common language for security aspects have culminated with what we would like to name, the most up-to-date ontology of security: the Common Criteria book. The general model, comprising the most important security concepts and the relationships between them, is depicted by Figure 2-1.

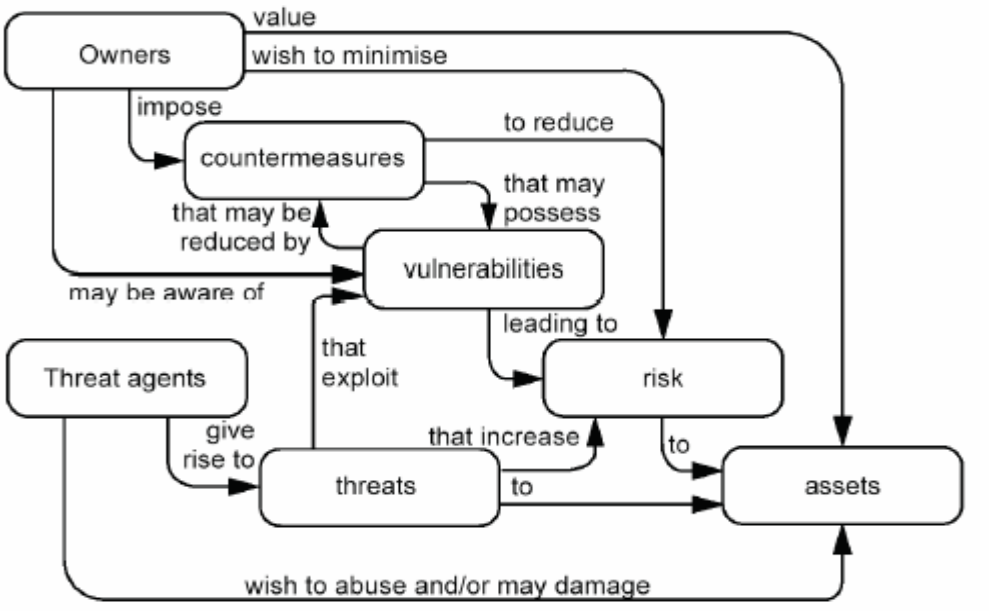


Figure 2-2: Security concepts and relationships [CC05]

[CC05] offers a thorough discussion around this picture. The overall explanation goes as follows. Security's main concern is protecting assets from threats. Although all threats are taken into account, the malicious threats are the ones mainly considered. The threat agents are the ones that are aimed at abusing the assets by creating threats that exploit the vulnerabilities of a system. The responsibility of protecting the assets belongs to their owners who place value on those assets and impose countermeasures that deal with the systems' vulnerabilities and meet the security policies. By examining the threats, the owners determine the risk associated with these and then select countermeasures to reduce those risks. By imposing countermeasures, the systems' vulnerabilities are reduced.

It is important for an owner to be aware of a countermeasure's validity. To this end, Figure 2-3 introduces the concept of countermeasure evaluation. In order to be sure of the validity of the countermeasures, the owners need to expose their assets to specified threats. But because the owners cannot be aware of all the aspects of a countermeasure, there is a need of the assurance of that countermeasure; that is, the extent to which a certain countermeasure can be trusted to reduce the threats. This way, the owners can decide whether to take the risk of exposing an asset to threats.

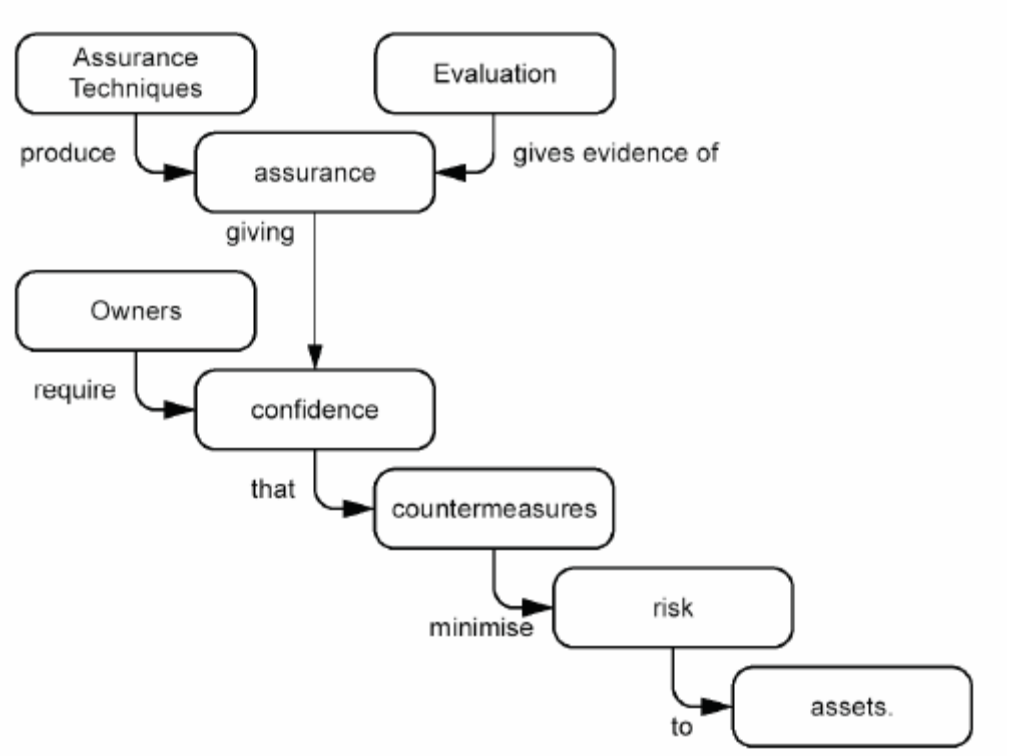


Figure 2-3 : Evaluation concepts and relationships [CC05]

In the end, it all comes to the security triangle since (CIA – see Section 2.5) when referring to security, one usually talks about Confidentiality, Integrity and Availability. Thus, breaking security means loss of confidentiality (inaccurate asset presentation to unauthorized recipients), loss of integrity (unauthorized modifications of the asset) and loss of availability (unauthorized removal of access to the asset).

Organizations are social systems
having as active elements or subjects the social individuals who behave
"according to assigned authority and corresponding responsibility
against a common background of social norms and values" [Die05a]

2.3 Enterprise ontology

An enterprise is primarily a social system [Die05a]. This means that the "active elements" of an enterprise are "social individuals". An organization observed as a social system is a consequence of people's purposes that originate in the "interpretations people make of the situations they find themselves in". Therefore, "organizations happen, and people act and interact in organizations, as a result of their interpretations." [Die05a]

The operating principle of a social system is "the ability of human beings to enter into and comply with commitments". Thus, organizations are social systems having as active elements or subjects the social individuals and these subjects behave "according to assigned authority and corresponding responsibility against a common background of social norms and values" [Die05a].

The ontology of a system is conceptually defined as the "understanding of its construction and operation" completely independent of the way the system is implemented. Operationally, the ontology of a system is "the 'highest' level constructional model" of that system. Thus, the ontology of an enterprise is defined as the "essence of the construction and operation of its organization" which is "completely independent of its implementation" [Die05a]. The core elements in the ontological model of an enterprise are actor roles, coordination acts/facts and production acts/facts.

The theory presented in [Die05a], defines the enterprise as a heterogeneous system composed as the "layered integration of three homogeneous systems" (Figure 2-4). All the three homogeneous systems belong to the category of social systems in the sense that the elements are subjects that enter and comply with commitments towards each other, it is only the production that differs (with regard to the type of production, we refer the reader to Figure 2-5: B stands for Business, I stands for Information and D stands for Data).

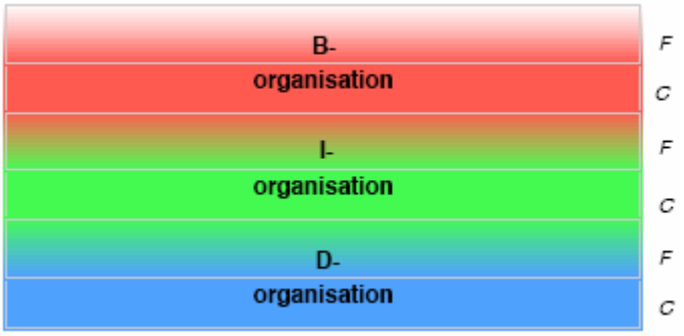


Figure 2-4: The layered integration of an enterprise

In addition to this, Figure 2-5 is important because it indicates that there is nothing above the B-organization of an enterprise: the B-organization is "the complete knowledge of the essence of the enterprise; all the rest in only realization and implementation" [Die05a].

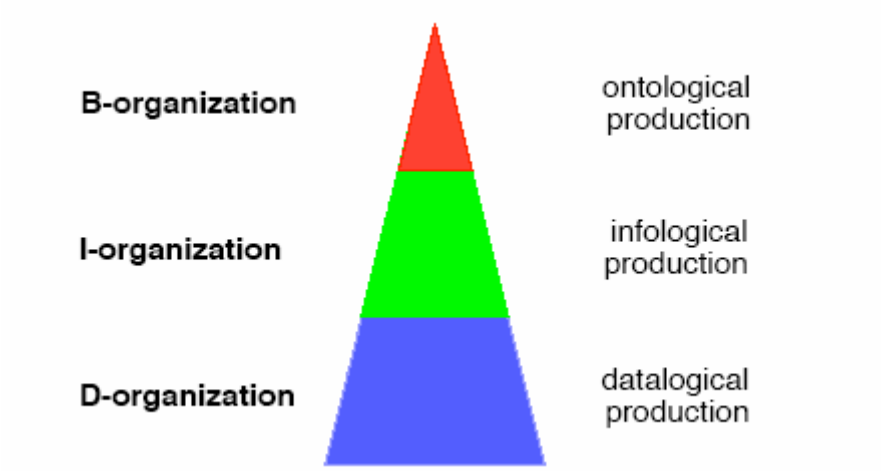


Figure 2-5: Representation of the organization theorem in DEMO [Die05a]

2.4 Summary of DEMO

This Section will familiarize the reader with the most important concepts incorporated within the DEMO methodology, namely the Ψ-theory about organizations and the way of working in DEMO.

DEMO is a methodology for (re)designing and (re)engineering organizations based on the *OER*[3] paradigm. It integrates in the class of modeling approaches from the Language/Aspect Perspective (LAP) together with methods like SAMPO, BAT, and Action Flow. The OER paradigm, the fact that its construction is based on two additional theoretical pillars (Stamper's Semiotic Ladder and Mario Bunge's Ontology), makes the main difference between DEMO and the former mentioned approaches [Die00]. In addition to these theories, DEMO also resides on general system theory and logic [Die04]. All combined, have generated the DEMO methodology based on Ψ- (to be read as PSI) theory about organizations.

Thus, in order to best portray DEMO, the reader is further acquainted with the idea of the Ψ- theory. The way of working with DEMO is then explained and discussed. For other, more detailed information about DEMO, the reader is referred to [demo], [Die99], [Die05a].

2.4.1 The theory behind DEMO – the Ψ (PSI) theory

In a DEMO context, an organization is a social system [Die05a]. In contrast to other modeling techniques that concentrate mainly on the function of the organization rather that on its construction and operation, DEMO presumes that in order to "(re-)design and (re-)engineer the business processes of an organization, one needs to have an understanding of the construction and action" [Die00]. Therefore, DEMO bases its approach on the white-box model instead of the black-box model [Die05a].

The PSI-paradigm (Performance in Social Interaction) is the theory DEMO is based on. This paradigm constitutes the essential difference between organizations and information systems. It states that "the operational principle of organizations is the ability of human beings to enter into and comply with commitments towards each other, collectively called social interaction".

The theory defines four axioms and a theorem:

1. The **operation axiom** states that an organization is a system of actors that perform production acts and coordination acts. By executing production acts (P-acts, e.g. transporting, selling), actors "contribute to achieving the purpose or the mission" of the

[3] OER stands for Order, Execution, Result

organization. By executing coordination acts (C-acts, e.g. requesting and promising), actors "enter into and comply with mutual commitments about P-acts".

An actor is defined as a person fulfilling an actor role and an actor role is defined as the "authority to perform a particular type of P-acts and the corresponding C-acts". DEMO states that authority is granted on the basis of competence and it has to be used with responsibility. Although the task of an actor can be fulfilled by, for example, computers, an actor always stays responsible for the outcome of an action [Die05a]. Hence, only human beings are able of "creating original new things"; artifacts like software systems or computers are not able of doing so. For a detailed analysis of the responsibility in DEMO, the reader is directed to Chapter 4.

2. The **transaction axiom** states that production and coordination facts occur in "generic socionomic patterns", named transactions.

The PSI theory defines 3 terms related to organizations: the atoms, the molecules and the 'fibers' of an organization. The 'atoms' are the P-facts and C-acts and they always occur in universal patterns - (business) transactions. Hence, the 'molecules' are the transactions. The 'fibers' are the business transactions which are a "tree-structure of connected transactions, starting with the transaction through which a service is delivered to the environment of the organization, or with an internal self-activation" [demo]. Hence, a business system is a "coherent structure of transactions" and a business process is a "succession of the statuses of these transactions" [Die05b]. The most important thing about a business process is that it is a "sequence of commitments between authorized and responsible social individuals" whose behavior is based on a "common background and social norms and values" [Die99].

According to the theory, a transaction has three phases: (1) the order phase (O-phase), when the two actors attempt to reach an agreement about the resulting P-fact (e.g. selling a good); the main coordination facts are request and promise; (2) the execution phase (E-phase), when the executor produces the result (e.g., deciding to sell); and (3) the result phase (R-phase), when the two actors attempt to reach agreement about the established result; the main coordination facts are state and accept. Figure 2-6 below depicts the basic transaction pattern.

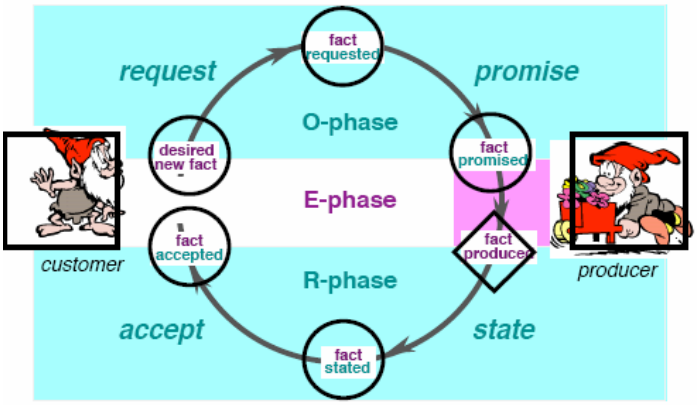


Figure 2-6: The generic transaction pattern [Die05a]

The structure of the standard transaction pattern is depicted in Figure 2-7. In addition to the generic transaction pattern, it includes also situations like declining a request, rejecting a statement or cancellation of coordination acts.

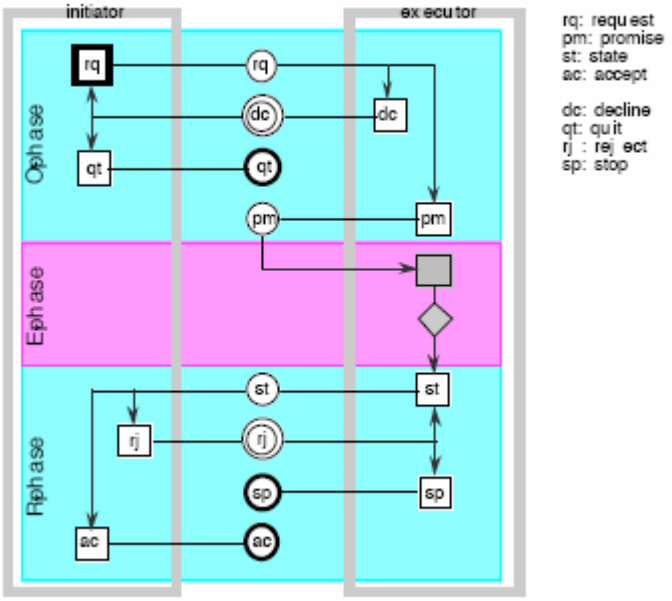


Figure 2-7: The standard pattern of a transaction [Die05a]

3. The **composition axiom** states that each transaction is either of the next three types: (1) embedded in some other transaction, (2) customer transaction or (3) a self-activating transaction.

4. The **distinction axiom** it is about the integrated role of human beings within an enterprise. Figure 2-8 depicts the summary of the axiom. According to this axiom, there are 3 basic human abilities: performa (the ability of human beings to produce original new things), informa (the intellectual capacity of human beings: reasoning, deriving new facts using the old ones) and forma (the ability of human begins to handle data and documents). By having this distinction, the complexity and diversity with regard to both coordination and production within an organization are achieved.

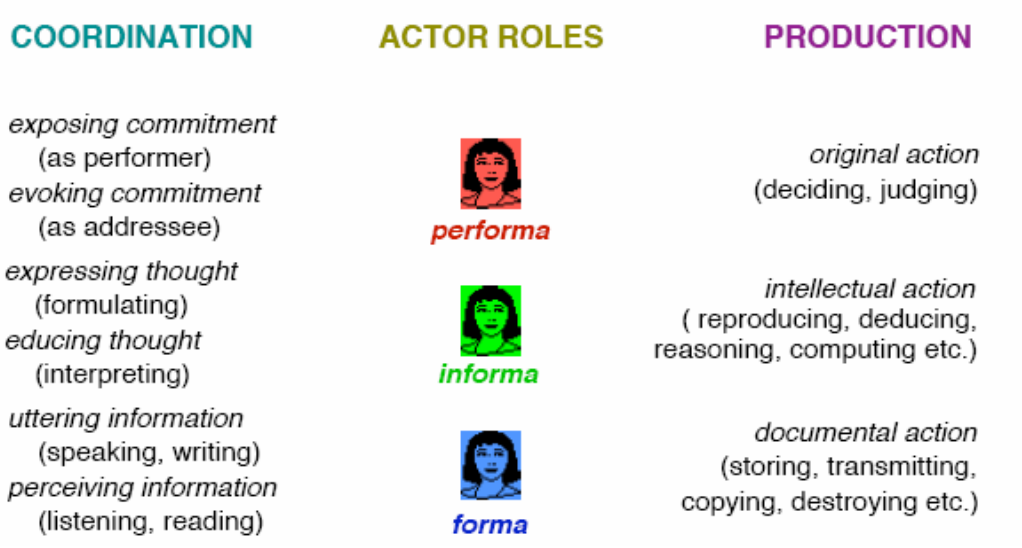


Figure 2-8: The distinction axiom [Die05a]

The **organization theorem**
The organization theorem takes the advantages and benefits offered by the above specified axioms and combines them into "one concise, comprehensive, coherent and consistent notion of enterprise, such that the (white-box) model of this notion of enterprise may rightly be called an enterprise ontology" [Die05a].

If we look back at the section where we presented the notion of enterprise ontology (Section 2.3), the reader can see the definition of the organization theorem (Figure 2-4 and Figure 2-5). Thus, the organization of an enterprise is the heterogeneous system composed out of three homogenous systems: B-organization, I-organization and D-organization, where the D-organization supports the I-organization and the I-organization supports the B-organization. The 3 systems are called the "aspect systems of the (total) organization of the enterprise" [Die05a] and there is nothing above the B-organization, thus "the knowledge of the B-organization of an enterprise is the complete knowledge of the essence of the enterprise; all the rest is merely realization and implementation" [Die05a].

The PSI theory is formalized and operationalized through the so-called CRISP model (we refer the reader to [Die05a] for a detailed description of the model). It constitutes the model necessary for the DEMO methodology.

2.4.2 The way of working in DEMO

As previously said, the ontology of an enterprise represents the construction and operation of its Business organization and the ontological model illustrates the essence of an enterprise "fully abstracted from realization and implementation issues" [Die05a]. It is important to be mentioned that contrary to other approaches, DEMO offers a solid connection between the essence of an enterprise and its implementation and realization. In DEMO, the ontological model consists if four aspect models each having its corresponding diagram. Figure 2-9 offers the overview of these models.

Practical experiences have shown that DEMO is mostly used in combination with other modeling approaches like UML and Petri Nets. For example, the Use Cases from UML can be derived from DEMO's Interaction Model, a DEMO transaction being the starting point for the functional specifications of a system [Mai02].

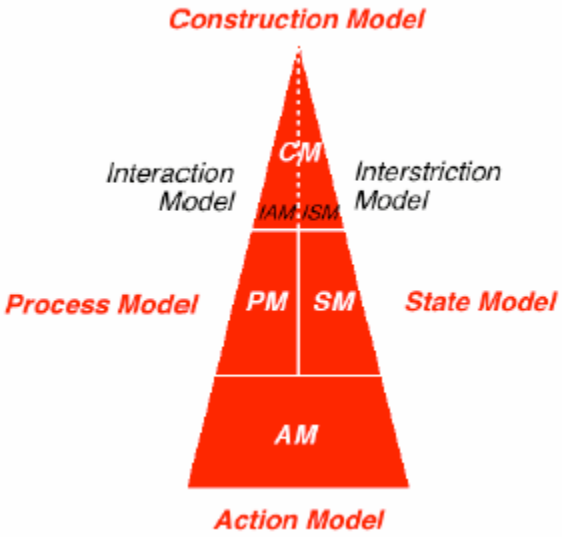


Figure 2-9: The ontological aspect models [Die05a]

The four aspect models will be further briefly described according to the theory in [Die05a]:

1. Construction model (CM)

The Construction Model it is the most concise type of model and "there is nothing above it". Together with the other three models, it constitutes the "complete ontological knowledge of an organization". The CM specifies the actor roles, the transaction types, the information banks (coordination and production) and the information links between the actor roles and the information sources. It can be divided into two models: the interaction model (IAM) and the interstriction model (ISM). The former depicts the "active influencing relationships" between the actor roles: who is initiating or executing a transaction. The latter pictures the "passive influencing relationships" between the actor roles: it restricts the actions among the actor roles by considering the facts produced by other actor roles. It is the compactness of the IAM that would make it so attractive to managers, because, for a large enterprise, it will offer them a compact but complete "enterprise map" on just an A1 page.

The CM's outputs are the Action Transaction Diagram, the Transaction Result Table (created together also with the State Model), the Actor Bank Diagram and the Bank Contents Table (created together with the State Model).

2. Process model (PM)

The Process Model identifies for each transaction type depicted in the CM, its specific transaction pattern. It also pictures the relationships (causal and conditional) between and within the transactions. It has as main output the Process Structure Diagram and together with the State Model, the Information Use Table.

3. Action model (AM)

The Action Model details the action rules that the actors use when dealing with their agenda; each agendum type has one or more action rules that are grouped based on the previously identified actor roles. The AM constitutes the bases for all other model types: it contains all the information that is also present in the CM, PM and SM, just in a more detailed and former manner.

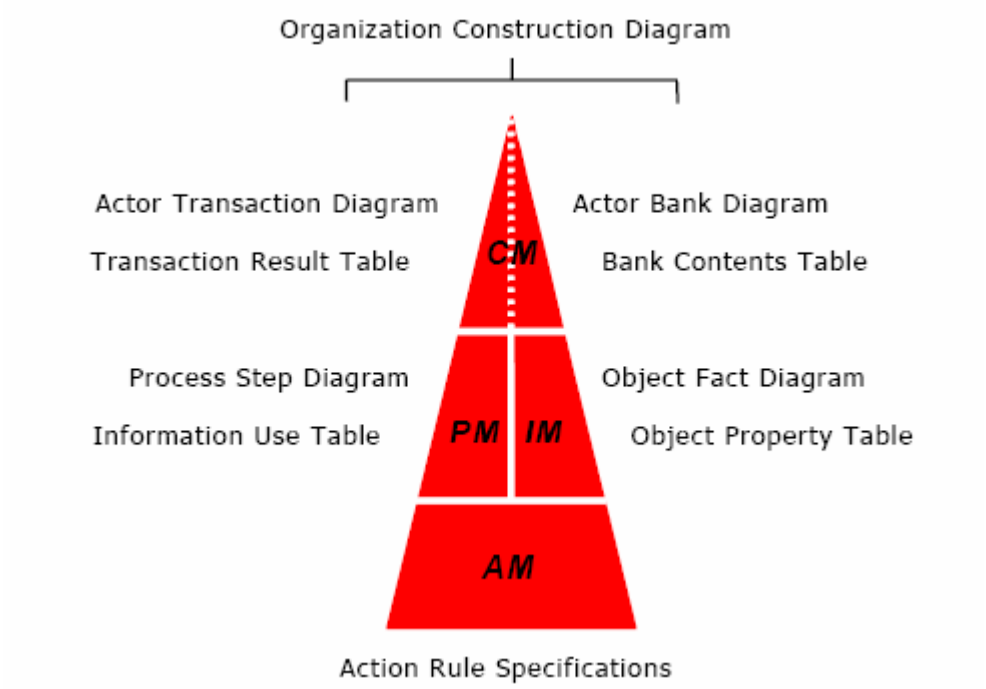


Figure 2-10: The aspect models and the corresponding diagrams

4. State model (SM)

The State Model displays the entity types and the fact types of the P-world and also the related laws and constrains, hence exhibiting the "*state space* of the P-world: all object classes, factum types and statum types, and the ontological coexistence rules". It has as main outputs the Object Fact Diagram and the Bank Contents Table.

As previously specified, the 4 aspect models offer the complete ontological knowledge of the organization. The way of working with the models and their corresponding diagrams is the anti-clockwise way as they are depicted in Figure 2-9. First, the method for identifying the actor roles and the boundary of the organization is applied. IAM is the immediate result expressed in the Actor Transaction Diagram and the Transaction List Table. Next follows the Process Model with the Process Structure Diagram for each transaction type and the action rules specifications (formulated in a pseudo-algorithmic language). The State Model is produced next and for the completion of the Process Model, also the Information Use Table is constructed. The Actor Bank Diagram and the Bank Contents Table form the ISM and complete the Construction Model. The Actor Bank Diagram together with the Actor Transaction Diagram constitute the Organization Construction Diagram. For a graphical depiction of the model types and their corresponding diagrams, we refer to Figure 2-10.

2.5 Summary and discussions

The first part of this Chapter has tried to answer the first research question of this thesis: to what extent does the current research in the field of security address and solve the information systems security concerns? We have this way identified that although research in information systems security approaches has gone towards a socio-organizational perspective, the approaches for managing security still reside on the idea of the "organization as a machine" [Dhi01b]. The main idea here, which also resides behind the DEMO theory, is that an organization is a social system. Responsibility analysis, the approach first undertaken in the third generation and continuing in the fourth [Sip05b], seems to be an appropriate starting point for analyzing security within DEMO. Ontology of security has been discussed and the Common Criteria have been identified as the most up-to-date ontology of security, comprising a complete, broad and general view of the most important security aspects and the relationships between them. Confidentiality - – Integrity - Availability (CIA) has been depicted as the most important security concepts, jointly called the security triangle (or triad).

Hence, we have identified that responsibility is a way of dealing with security of interest to us. Therefore, we think that it is important to portray the view on security as seen by some of the promulgators of the idea of '*responsibility - a security issue*'. To this end, G. Dhillon presents an additional opinion on information security management in the new millennium [Dhi01a]. The author states that in addition to the CIA triangle of security there should also be added the social extent, the RITE (responsibility, integrity, trust and ethicality) rectangle (Figure 2-11):

Figure 2-11: Information Security Principles in the New Millennium

- **Responsibility** (and knowledge of roles)
  It is recognized that in the modern enterprise, empowerment seems to be a more effective way of management than the traditional vertical management organization. The responsibility in the modern enterprise goes beyond the accountability – blame attribution structure. It needs people that are capable of taking responsibility on an ah-hoc basis, that are able "to take a burden unsolicited", to understand what "their responsibilities should be". This approach to responsibility takes a little bit different path than the way responsibility and areas of responsibility are tackled within DEMO (Section 4.2 and 4.3). The Process Model in DEMO defines all the steps an actor role has to and is allowed to take. The responsibility in RITE goes towards a more sensible, organizational-like issue, because responsibility refers not only to what one can do but it refers to the "handling the development of events in the future in a particular sphere" [Dhi01a].
- **Integrity** (as requirement of membership)
  It is a widely known fact that a large number of security threats of organizations come from inside it, from its employees, mainly from disclosure of business related information. Therefore, a person's integrity is essential.
- **Trust** (as distinct from control)
  The modern enterprise is based less on command and supervision techniques and more on self control and responsibility. An interesting concept mentioned is the half-life of trust. It refers to the fact that trust within a certain circle (members of a community, of a project, of an organization) lasts only for some time after a certain physical, face-to-face meeting has taken place. After some time, another meeting needs to be in order for trust to be established between the members.
- **Ethicality** (as opposed to rules)
  Ethical behavior refers to those situations within an organization when one has no clear rules of behavior, no formal procedures to follow. It is about "the ethical content of informal norms of behavior" [Die01a]: how one should direct its actions in dynamic, not previously defined, situations. Ethicality is important because the right/secure behavior cannot be further governed only by rules as the modern organization has as one important characteristic its dynamicity.

The second part of this Chapter has further introduced the theory about DEMO and enterprise ontology. This way, we have seen that DEMO also encloses and explains the aspects of responsibility within an enterprise: organizations are social systems having as active elements or subjects the social individuals who behave "according to assigned authority and corresponding responsibility against a common background of social norms and values" [Die05]. Hence, as an answer to the second research question of this thesis, we can now state that **responsibility is the thread connecting security with DEMO**. Therefore, the report will mainly consider (Chapter 3) responsibility analysis approaches to security when trying to address security at a DEMO level.

Responsibility - a key issue for security [Str93]

3 Responsibility modeling – related work

The second Chapter of this report has identified responsibility as the connection thread between DEMO and security. Therefore, how is responsibility defined and which is its role in dealing with security within an organization are questions that will be dealt with along this Chapter. This way, the third research question will be answered. Given all these, security approaches based on the idea of responsibility will be further described. In the end, the main findings will be summarized and discussed.

There are mainly three responsibility modeling approaches of information systems security identified in the literature [Dhi01a]: (1) Dobson's approach from 1993 [Str03], (2) Thomas and Sandhu in 1994 [Tho94], and (3) Backhouse and Dhillon in 1996 [Bac96]. To these three, there are two more that, although they do not start directly from responsibility, can be integrated in the same line of thinking. These are (4) McDermott and Fox in 1999 [Mcd99] and (5) i* [Liu03a]. The idea proposed by the last two starts from identifying the actors of the system in terms of their capabilities which used in a malicious manner, can breach the security of the system. We will further take all these approaches and try to constructively identify and discuss their relations with this thesis' objective.

3.1 Responsibility modeling drives security

The authors of [Str93] define responsibility as the "relationship between two agents regarding a specific state of affairs, such as the holder of the responsibility is responsible to the giver of responsibility, the responsibility principal" (Figure 3-1). The authors advocate that security requirements reside in the idea of responsibility in the sense that "the responsibility holder needs to do things, needs to know things and needs to record things for subsequent audit".

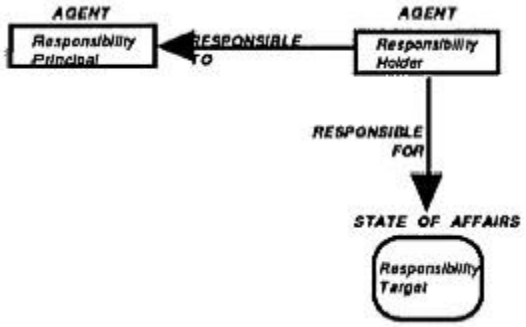


Figure 3-1: Responsibility relationship between 2 agents [Str93]

According to its definition as given in [Str93], responsibility consists of:

a) who is responsible to whom
b) the state of affairs for which the responsibility is held
c) a list of obligations held by the responsibility holder (indicating how the responsibility can be fulfilled)
d) the type of responsibility (i.e. accountability, blameworthiness, legal liability)

Through obligation, the authors refer to the actual job that needs to be performed; an activity is the description of how an agent has to perform something; and the responsibility is expressed as the reason one would do something. The main difference between the first and the last is that a responsibility is for a state of affairs, whereas an obligation is "to do something that will change or maintain that state of affairs". Hence, the obligations actually define how a particular responsibility is fulfilled [Str93].

An important concept in the world of responsibility is the process of transfer or delegation. The essential feature about responsibilities and obligations in Dobson's approach is that only the latter can be transferred. When an obligation is delegated, the agent that delegates becomes the responsibility principal, whereas the one that receives the obligation becomes the responsibility holder (Figure 3-2).

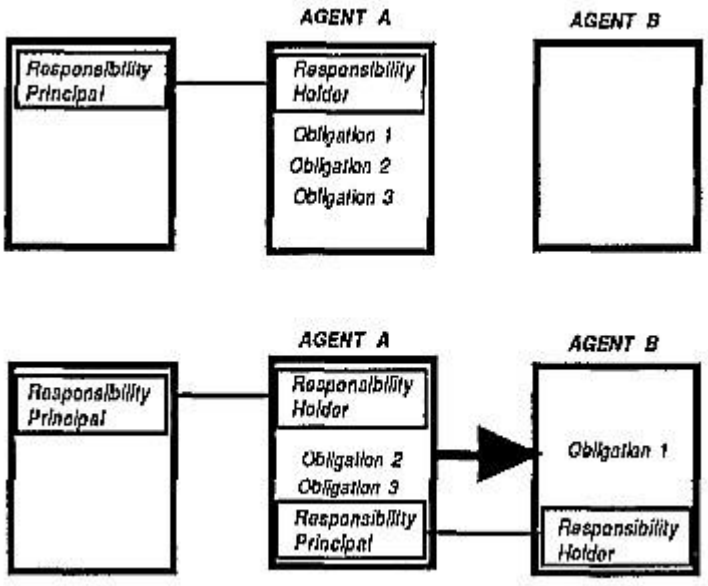


Figure 3-2: A responsibility relationship created by the transfer of obligation [Str93]

According to the work in [Str93], there are 3 kinds of security requirements that can be withdrawn from responsibilities: 'a need-to-know' list for information, a 'need-to-do' list for action and a 'need-for-audit' for recording a history. By inferring these lists, the framework demonstrates how security policies are actually derived. The 'need-to-know' actually refers to classical security policy which states that users need to know things in order to perform their tasks, so they should be granted access to information, but also states which information should be kept away from one's attention and access possibility. The 'need-to-do' and a 'need-for-audit' refer to the things one needs to do and register for the solving of a possible subsequent accountability problem.

Some years later, the DIRC project continued on the same line by widening and improving the idea of responsibility as a "key issue for security" [Str93]. Dobson and the DIRC team are currently trying to define and refine the necessary concepts and notations [dirc].

In their opinion, there are 2 distinct types of responsibilities: (1) consequential responsibility (who takes the blame for something) and (2) causal responsibility (who makes something happen). Their argumentation is based on the distinction between social individuals and machines. Whereas the latter can be casually responsible for assuring that some state is conserved, it is the human that is consequentially responsible if the respective machine fails to fulfil its task [Dob05]. The group considers that these concepts "will help system designers identify system vulnerabilities and provide a basis for supporting recovery from failure".

The group's work sustains that in order to correctly identify the responsibilities structures within an organization, ethnographic research needs to be performed because differences have been identified between (1) what people do, (2) what people say they are doing and (3) what managers think that people are doing. Hence, responsibilities cannot be simply observed, they need "to be inferred and constructed". They take the Soft Systems Methodology approach to identifying weaknesses in an organization; this is, the construction of a normative model that identifies which are the characteristics of a certain type of organization. The idea is to take this

normative model and compare it with a descriptive model of the real organization; the differences and week points can be identified and further discuss. The DIRC project constructs a normative model based on responsibilities and the descriptive model based on ethnographic observations [dirc].

Another approach based on responsibilities and obligations is the one described in [Tho94]. The authors' focus is mainly on the "provision and maintenance of integrity" in computerized information systems, with particular emphasis on integrity from an enterprise standpoint. The authors concentrate on responsibilities and authorization structures when they try to build a better authorization model. The paper concentrates mainly on technical issues and the main design objective is intended towards technical systems. This is the main reason that makes this approach not suitable for the purposes of this thesis. A second reason would be the fact that the work described is mostly conceptual and a formal model does not exist.

3.2 Responsibility modeling and organizational semiotics

The second Chapter of this thesis has brought to the reader's attention that there have been some approaches of using semiotics within the area of security: i.e. the work in [Bac96] uses semiotics to "interpret the security implications of organizational actions". We will further get the reader acquainted with what semiotics and organizational semiotics are about and describe a security framework based on a semiotic approach as it is depicted in [Bak96].

3.2.1 Organizational semiotics – definition and concepts

Probably, the most used definition of semiotics is: "the study of signs, both individually and grouped in sign systems, and includes the study of how meaning is made and understood" [wik]. From the fields of semiotics, organizational semiotics stands out as the science that looks at an organization and tries to understand it based on "the use of signs, texts, documents, sign-based artefacts and communication, thereby using the results of for instance psychology, economics, and information systems science as basic disciplines" [Gaz04]. In a more formal context, organizational semiotics "is an emergent discipline to study the nature, characteristics, function and effect of information and communication within organizational contexts" [org].

There are three major approaches that belong to organizational semiotics; these are system-oriented approaches, behaviour-oriented approaches, and knowledge-oriented approaches [Gaz04]. Within the second approach two directions can be identified: information field based organizational semiotics and interaction structure based organizational semiotics. From the two, the former is of importance to us. Information field based organizational semiotics or the Stamper school of organizational semiotics [Gaz04] considers organizational semiotics as a tool to develop and improve organizations. Relevant literature is [Liu00] and [Sta73].

[Sta73] has proposed a set of methods that facilitate the study of the use of signs in organizations and their social effects. The most important methods that belong to organizational semiotics are: Semantic Analysis and Norm Analysis. Semantics Analysis "delineates the area of concern of an organization and identifies the basic patterns of behavior (affordances) of the agents". Norm Analysis "describes how an agent can judge a situation and take actions" and a norm "describes the responsibilities and authorities of each member, and establishes regularities of behavior" [Liu02].

3.2.2 A semiotic framework for responsibility

The line of thinking introduced in [Dhi96] points out the idea that analyzing the structures of responsibility in an organization would lead to the development of secure information systems. The approach is an interpretative one and presents a "framework for security evaluation based on ontological interpretation". The work performed is similar to and considered to be a continuation of Dobson's ideas as they were introduced in [Stre93] as they both envision the organization as a framework of responsibilities.

The authors of [Dhi96] try to analyze an organization as "patterns of behavior" and in doing so they start from responsibility, more precisely from patterns of responsibility. Their ontological-epistemological approach is based on the work in [Sta88]; that is, understanding "social norms and individual affordances" as "patterns of behavior at social and individual levels". Their framework can be pictured as a semantic schema (Figure 3-3).

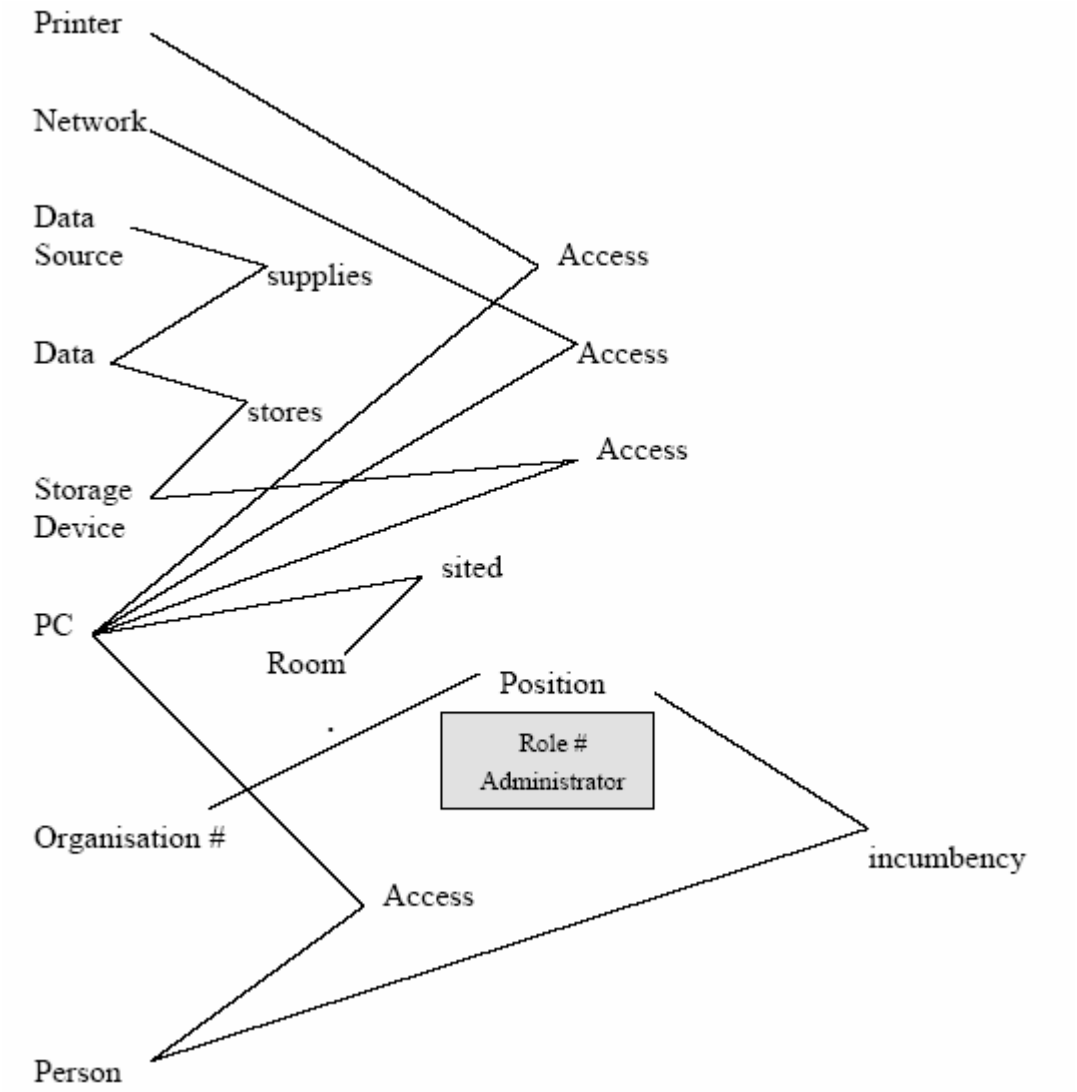


Figure 3-3: A semantic schema for secure environment [Bac96]

The nodes in the chart portray the invariants as patterns of behavior to be realized by agents. The symbol # denotes the sign for individualism and the ontological dependency means that the entities/nodes on the right can only be realized after the ones on the left side have been recognized (e.g. the *Person* and the *PC* must both exist if there is to be any access from the first to the second).

The framework presented is a useful platform to study the norms and the structure of an organization [Dhi96]. The schema does not present rules or specifications, but only the structural constraints that restrict the behavior of agents. It defines the agents or people that decide at each node: who should access *this* computer (the entity PC in Figure 3-3), what data should be accessible from *this* computer, in which room *this* computer should be placed. This way, the

object system is better understood by the one that analyses the system and it helps at identifying issues like attribution of blame, accountability and authority [Dhi06].

3.3 'Abuse cases' for security requirements

The authors in [Mcd99] came with the idea that by modifying the well-known use-cases from UML, they constitute the basis for security requirements. They adapt use cases in abuse cases to analyze the security requirements in a simple way that is accessible for non-security specialists. For example, in Figure 3-4, the hacker is a system abuser that can tamper with information or have access to information that he is not allowed to.

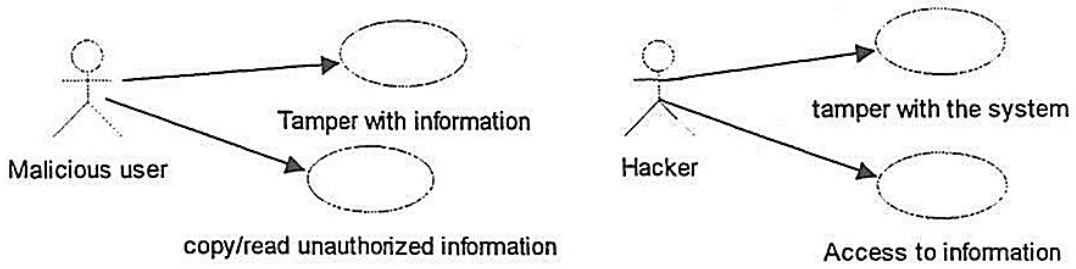


Figure 3-4: Abuse cases by McDermott & Fox [Dhi01a]

With regard to this thesis purpose, the method is not a very relevant one due to the use of UML used in the area of software engineering. In this regard, another work that uses a similar idea with the one of abuse cases, but formulated from a social perspective, is the one presented in [Liu03a].

The authors in [Liu03a] claim that due to the fact that security and privacy issues are a result of human "concerns and intents", they should be modeled through social concepts. To accomplish this, the authors propose a methodological framework for analyzing security and privacy based on the concept of strategic social actors, the so-called i* framework. In doing so, they analyze each actor implicated in the system. An actor is differentiated in: role (encloses responsibilities), agent (encloses capabilities and functionalities; it can be a human or a machine), and position (a set of roles grouped together and assigned to an agent). After identifying and analyzing each actor, its goals and the dependencies between the actors, the framework proposes the transformation of each actor into an attacker of the system - an approach founded on the basic principle "guilty until proven innocent". This way, by using its capabilities, each actor can pose a/more threats to the system. After analyzing the threats and, this way, the vulnerabilities of the system, a countermeasure analysis is performed in order to evaluate which are the measures to be taken in order for the system to be a secure one (Figure 3-5).

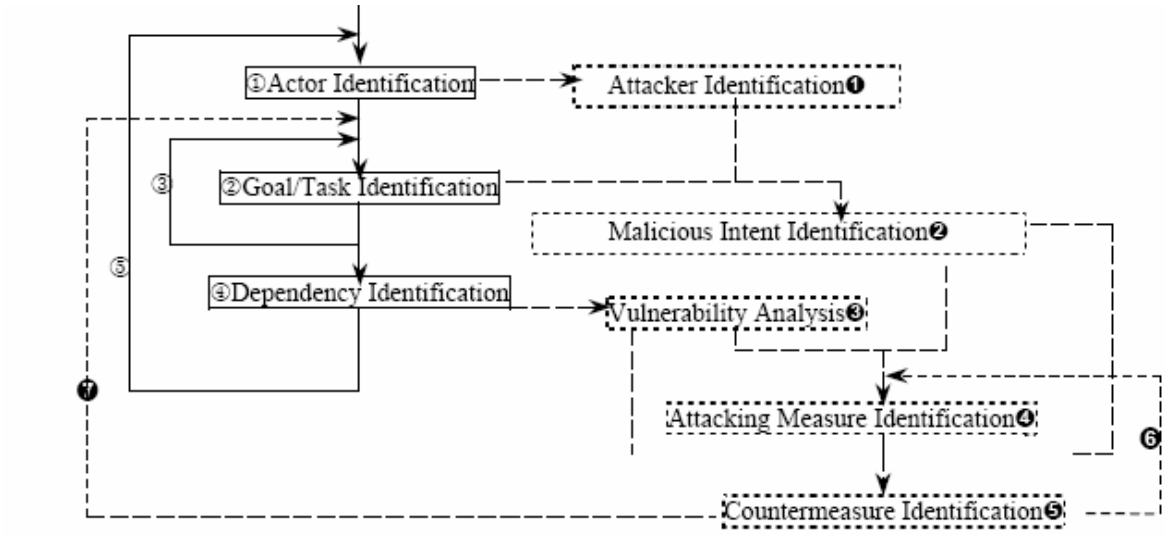


Figure 3-5: Requirements elicitation process within i* [Liu03a]

The main resemblance with the DEMO approach resides in the fact that the authors admit that a software agent can pose this kind of threats to a system only if manipulated by a human agent. This is because the latter is the one that can perform things it should not perform (thus 'new' things) and eventually make a software agent act in an improper way; a software agent cannot do so by itself. Thus this idea can be 'translated' in the DEMO expression: only humans can create "new and original" facts. Although the framework uses relevant ideas starting from roles which enclose responsibilities, the approach does not have a very strong theoretical foundation. But the idea of the considering actors as possible attackers is a sound concept when determining the vulnerabilities of a social system.

3.4 Summary and discussions

This Chapter has tried to make the reader familiar with the responsibility related theories that are necessary for the conduct of the research. To this end, the responsibility approaches to security present in the current literature have been identified and described.

As a better framework for the discussions in this Chapter, we have chosen Table 3-1 to summarize and discuss the previously descried responsibility approaches to security. The table presents each approach with a short description and the strengths and weaknesses of the underlying method in relation to this thesis' purpose.

(1) The approach	(2) Objectives	(3) Strengths	(4) Weaknesses
Dobson's approach from 1993 [Str03] and the continuation : the later DIRC project	Defines the responsibilities and obligations within an organization, concentrating on building a 'need-to-know', 'need-to-do', 'need-to-audit' list for information. This way, the authors show the path to the foundation of security policies. It also identifies delegation as the process that triggers the vulnerable points of the analyzed system.	It argues that responsibility is a key issue of security. The responsibility-obligation framework in [Sre93] and the process of delegation are well defined and argued.	[Str93] only covers the analysis of security requirements and does not include vulnerability and a countermeasure analysis. In addition to this, the example presented is weak and without convincing practical evidence.
Thomas and Sandhu in 1994 [Tho94]	Concentrates on responsibilities and authorization structures to build a better authorization model from an enterprise perspective.	By analyzing the responsibilities structures within an enterprise, the authors demonstrate that these lead to access control, thus authorization.	The main design objective is intended towards technical systems, whereas our attention is towards social systems. In terms of security analysis, it presents the disadvantage that its only concern is the authorization.
Backhouse and Dhillon in 1996 [Bac96]	Reasons for the idea that analyzing the structures of responsibility in an organization would lead to the development of secure information systems. The approach is an interpretive one and presents a "framework for security evaluation based on ontological interpretation".	The recognition and verification of the fact that responsibility leads to security design.	Because of the new notations that are completely different from other approaches, it would be difficult to integrate the proposed framework with methods in the category of information systems design [Dhi01a].
McDermott and Fox in 1999 [Mcd99]	Adapts use cases in abuse cases to analyze the security requirements in a simple way that is accessible for non security specialists.	It introduces the concept of abuse cases when analyzing the vulnerabilities of a system.	Usage of UML that is suitable for software engineering whereas we are interested in enterprise engineering.

(1)	(2)	(3)	(4)
The i* approach [Liu03a]	Claims that due to the fact that security and privacy issues are a result of human "concerns and intents", they should be modeled through social concepts. To accomplish this, the authors propose a methodological framework for analyzing security and privacy based on the concept of strategic social actors.	Positions the idea of abuse cases into a social context and this way helps at identifying the potential abusers, their malicious intents and the vulnerabilities of the analyzed system. It also presents the advantage of introducing a countermeasure analysis (none of the previous approaches did).	The framework proposed is based on i*, and agent-oriented modeling language. According to this, the actors can be as well human agents and software agents. Although the difference between the two is recognized, the framework does not clearly distinguish between them. This is a main disadvantage when we think that in DEMO, organizations are "social systems having as active elements the social individuals".

Table 3-1: Responsibility based approaches to security

As we see from the above table, all the presented approaches to security start from the idea of responsibility (the last two do not directly specify responsibility, but the actors include responsibilities – the approaches are similar to the others due to this characteristic). All methods have their strengths and weaknesses with regard to our purpose and in order to establish what from these approaches could be suitable for an analysis of security at a DEMO level, we need to analyze DEMO from the same perspective. Therefore, in the next Chapter, a critical analysis of DEMO in terms of responsibility will be conducted. After we would have done this, the approaches described and analyzed in this Chapter will be compared with the DEMO approach to responsibility and based on this comparison, a possible method for analyzing security starting from DEMO will be proposed. For demonstration purposes, two case studies will be analyzed related to this proposed method.

Responsibility – authority – competence
"No responsibility without authority:
authority and responsibility are as the two sides of a medal"

4 Analysis of DEMO from a Security perspective

We have previously pinpointed responsibility as being the thread connecting DEMO with security. Therefore, in this Chapter we will try to identify and discuss all the aspects related to responsibility encapsulated by DEMO. We will talk about how DEMO defines the concepts of actors and actor roles; about the relationship responsibility – authority – competence; about how DEMO identifies the areas of responsibilities of actor roles and about the process of delegation in DEMO. The findings will be then summarized and the DEMO approach to responsibility will be compared with the previous analyzed security modeling approaches based on responsibility (see Chapter 3). At the end, a possible method for analyzing security starting from DEMO will be proposed.

4.1 Actors and actor roles

The ontological model of an enterprise defines the actor roles, the coordination acts/facts and the production acts/facts as the enterprise's core elements. According to the operation axiom of the PSI theory (see Section 2.4.1), an elementary actor role is the "atomic amount of authority and responsibility" that acts to fulfill the operation of the enterprise. In doing so, an actor role has the authority to perform a certain type pf P-acts and the corresponding C-acts. A transaction is a succession of coordination acts and it is the actor role that stands as the initiator and executor of a transaction. A transaction always involves 2 actor roles that work together to attain a specific result. The elementary actor role it is the producer of just one transaction and the customer of none, one or more transactions. [Die05a]

An actor is "is a subject in its fulfilment of an actor role". Actors are the only active elements of the enterprise. Other elements such as artifacts are not considered to be active elements but only supporting actors and they cannot replace the subject, the social individual. Actors act autonomously and their actions are not initiated by events since this is too mechanical. They base their actions on a reason. Each actor is committed to deal with an agenda; this is, a collection of C-facts, each with a proposed time for action, and each guided by specific action rules. As a result of their actions, the actors perform one or more C-acts. These C-acts are intended for other actors meaning that the resulting C-facts will be added to the agenda of the latter category. In this way, actors permanently provide each other with work, with facts to deal with; the exception to this rule is represented by the terminal C-facts. [Die05a]

The concept of roles, namely role-based access control, plays an import role when controlling the access to the resources of an enterprise [Jur05]. Role-based access control is a security concept important when trying to manage the permissions in a system, especially when the actors fulfilling different actor roles change very often. In this case, the rights and permissions are assigned to roles (actor roles in DEMO) and not to users (actors in DEMO). The latter will 'inherit' the rights and obligations appointed to the roles they perform within an organization. Although role-based access control proves to be an efficient way of managing information/resources access, it is to be noted that does not suffice when we think, for example, at security in the health environment. For instance, only one doctor, John, is allowed to see the medical data of a certain patient, Lara. Therefore, it would not be a good solution to assign responsibilities, thus give authorization rights, for the actor role 'doctor' to have access to all the data of the actor role 'patient'. This issue is in direct relation to the concept of fulfillment of actor roles in DEMO, which will be further described.

A DEMO actor role can be fulfilled by one or more subjects and there are three different ways of doing so [Die05a]:

1. sequentially: it refers to the situation when an actor role is fulfilled by two or more subjects one after another (e.g. a guarding shift);
2. concurrently: it refers to the situation when an actor role is fulfilled by different subjects that complete the same kind of work (e.g. all persons within an enterprise that fulfill the same actor role, like salespersons in a car-producer enterprise or the nurses in an hospital);
3. collectively: it refers to the situation when an actor role is fulfilled by different subject but in the same time, collectively. This way, the group has together the authority to perform specific acts (e.g. one or more doctors together with one or more nurses have together the authority to perform a certain operation of a certain type and at a certain point in time).

The process of fulfillment of actor roles is similar the process of delegation (Section 4.4). It will then become even clearer how this can affect on the security of the entire system. For now, we will try to explain how the fulfillment of actor roles is connected to security and later on, the process of delegation will also be described.

A first issue has to do with the fact that more than one subject fulfilling an actor role drives to problems of assigning blame for people for the actions they perform (accountability). Thus, by having more that one actor fulfilling the same actor role, the system becomes vulnerable. For example, if we take the sequential type of fulfillment: if one doctor in a hospital has a shift from 9 to 17 and the responsibility to take care of a certain number of patients, all the acts that he performs need to be recorded in such a way that the doctor that comes after him from 17 to 1, having the same kind of responsibilities (thus fulfilling the same actor role) knows all the things he needs to know for the good implementation of his responsibility. Such a record becomes extremely important when assigning blame if some things go wrong. By having a record with all what happens and the respective times of action, the 'guilty' person can be correctly identified and penalized accordingly. In terms of security, this has to do with audit. Thus, the fulfillment of actor roles as described by DEMO is a first and not the only one that would drive towards a security policy for audit within an enterprise. If related to the other responsibility modeling techniques, this represents the "need - to - audit" list as it was described in [Str93].

A second issue has to do with the fact that role-based access control is not always sufficient for insuring the security of a system (i.e. the case of the actor role doctor and actor role patient explained above) and special additional rules need to be applied to prevent unauthorized disclosure of information in such cases.

Although assigning subjects to actor roles it is a problem of implementation thus it is abstracted from the ontological model, we consider this idea of separation of fulfillment of actor roles as the bullet that triggers the need for establishing a security policy in regard to audit within an enterprise.

4.2 Responsibility – authority – competence

The Construction Model, the Interaction Action Model specifically, is where are first depicted the ontological units of responsibility, authorization and competence. It is later on, when constructing the process diagrams that the actual areas of responsibility are defined. But first, it is the PSI theory that addresses this issue.

According to the operation axiom from the PSI theory, an organization is a system of actors that perform production acts (Figure 4-1, the plain arrow from the actors' square to the P-world's diamond) and coordination acts (Figure 4-1, the plain arrow from the actors' square to the C-world's circle). The dashed arrows in Figure 4-1 refer to the fact that actors take into account both C-world and P-world in their actions. Figure 4-1 also depicts the concepts of responsibility, authority and competence, explaining in fact what drives an actor to complete its actions [Die05a].

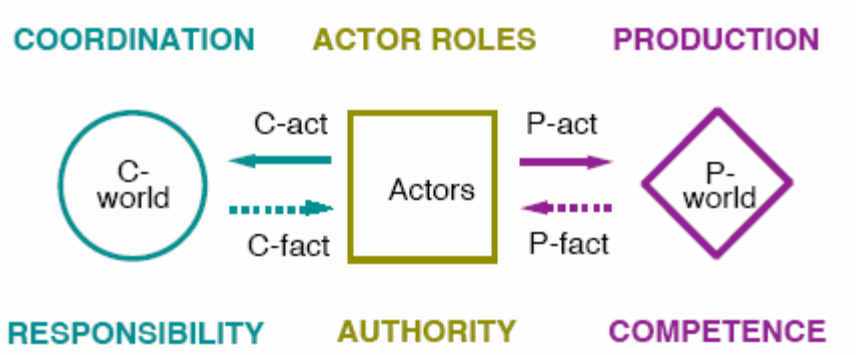


Figure 4-1: Responsibility, authority and competence in DEMO [Die05a]

- Competence: is "the primarily manifest in production"; it represents the ability of a subject to perform certain P-acts and their corresponding C-acts (e.g. to be able to practice ones profession, a surgeon, who has the knowledge and experience necessary for his profession , needs to be appointed by a hospital, or another medical body);
- Authority: when executing ones competences, one acquires the authority to act on behalf of the institution that appointed him (e.g. the above mentioned surgeon has the authority to perform operations in behalf of hospital X that appointed him);
- Responsibility: it is "primarily manifest in coordination"; as a member of an organization, one is expected to use ones authority in a responsible manner (e.g. the surgeon is expected to act responsible towards the clients/patients of the hospital that appointed him with the authority of acting).

It is another older story presented in [Die06] that makes the connection between the three ontological units more clearly: *It is certain that one could give in principle all red gnomes (subjects) authority for everything, but this seems not such a good idea because not everyone is for everything competent. We like to handle authority carefully and connected to responsibility, and make sure that everyone has the competence that belongs to its responsibility. A red gnome (subject) is accountable for the way in which he has used his authority. Authority and responsibility are as the two sides of a medal*[4].

Thus there is no responsibility without authority and by having the authority to do something, a subject can be held responsible for its actions. And by having the responsibility to perform certain acts, one becomes accountable for those acts. The next section extends the concept of responsibility and talks about the way DEMO defines the areas of responsibility and how by defining these areas, a point for establishing a security policy is triggered.

4.3 Areas of responsibility

The Process Model defines the areas of responsibility of each actor role. For each transaction type, all the steps to be performed are depicted by the Process Structure Diagram. As a result, the steps that are not depicted are not allowed to be performed. A process step is actually a transition in the C-world and it is represented by the C-fact and the corresponding C-act. Thus, for a specific transaction, the PM defines all the C-acts that a particular actor role is allowed to perform. In addition to this, the PM also defines all the coordination and production banks that an actor role needs information from in order to act correctly (Figure 4-2) [Die05a].

The areas of responsibility are rigorously defined in DEMO: nothing is omitted and nothing is redundant or unnecessary. Therefore, we can state that the PM actually illustrates all

[4] adapted translation from the Dutch language, with the help of Rob Weemhoff

the acts an actor role is allowed and has to perform, thus all its capabilities in that role. It also illustrates all the information a particular actor role needs to know in order to exercise its responsibility. Thus, starting from Process Model, the 'need-to-do' and 'need-to-know' lists that are mentioned in [Str93] can be constructed. They constitute the starting point for the foundation of the classical security policy (users need to know things in order to perform their tasks, so they should be granted access to information).

It is finally the interstriction model that portrays the overall picture of the 'need-to-know' list by drawing the so-called interstriction links between the actor roles and information banks (these links are passive influences, since they are not actively influencing the actor roles). They express what an actor role is allowed and needs to know in order to properly execute its action rules. Consequently, by not drawing a link from one actor role to a particular information bank, the model reveals which information should be kept away from one's access possibility. This way, we can derive the authorization rules about who should be granted access to certain information and who should not (see Figure 4-2): "the issue of ownership of data is made fully transparent by the Construction Model" [Die05a].

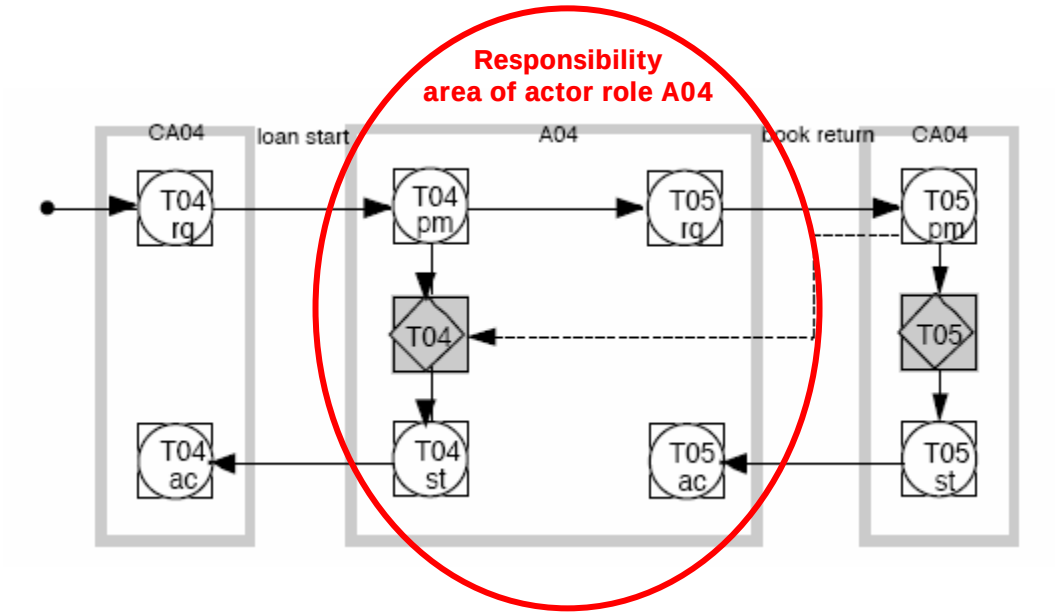


Figure 4-2: Process Structure Diagram for the process 2 of the library: area of responsibility of actor role A04 [Die05a]

As stated just two paragraphs before, the PM presents all the acts an actor role is allowed and has to perform, thus all its capabilities. A problem might appear when the subject fulfilling that actor role is using its capabilities in a malicious manner; this is, he is trying to use its capabilities in a way that would harm the system and its good functioning. The idea comes from the methodology used in [Liu03a] which envisions that all actors can use their capabilities in a detrimental manner to the system and this way, they can harm the system. To put this idea in a DEMO language, it would mean to suppose that all the subjects fulfilling an actor role use their capabilities to act in a way they should not. The intention is to come with a strategy that would predict things like this to happen.

A first idea for such a strategy resides in the way the Process Model is defined. The process model states that only the steps that are described in the PSD can be performed and any step that is not portrayed cannot be performed. By asserting this, the Process Model presents a

kind of protection of the system from malevolent users/abusers: for the situation in which one subject fulfilling an actor role might be tempted to do something he is not allowed.

But this countermeasure does not come with a solution when a subject deliberately does not perform an act as he should. The actors fulfilling different actor roles can tamper with the information they have access to by, for example, disclosure to third parties. Also, there is the case when data are introduced in the production banks in an erroneous manner (intentional or unintentional): this affects the activities of all the other actor roles that depend on the accuracy of the data they use in performing their roles. Thus, the way the actors depend one on each other it is important since the point of dependency represent a vulnerable point in the entire system.

So, it can be concluded that the areas of responsibility represent the points from which the security policy for the 'need-to-know' and 'need-to-do' can be derived. The dependencies between the actor roles represent the vulnerable points of the system since an actor fulfilling an actor role in a faulty manner affects the actions of all the other actors that depend on him. The concern is to find possible ways that deal with these vulnerabilities.

4.4 The process of delegation

It has been previously mentioned that delegating responsibility may impose vulnerabilities to the system. In direct connection with the areas of responsibility as they were explained in the previous section, the delegation process is also depicted by DEMO. The chosen method of explanation is by example and it has to do with the fulfillment of actor roles (thus implementation issue) within the same transaction.

In DEMO, an actor that performs the promise in a transaction is considered to be the authorized person for the execution of a transaction. Hence, the actor that promises and executes a transaction, thus authorized to do so, has the full responsibility for the correct implementation of that transaction. It can be the case that one actor that has the authority to do something needs to transfer that authority to another person. This process is called delegation and it means that the authorized person will transfer the authority to do something to the delegated person but not the responsibility (the former remains responsible). The process of delegation, as the one concerning the fulfillment of actor roles, is an implementation issue in DEMO.

The example of delegation presented in [Die05a] is depicted in Figure 4-3. It is taken from the classical Pizzeria case of DEMO. It appears in the situation when an actor role needs to be fulfilled by more than one actor in the same transaction. It goes as following: the 'order taker' is authorized for fulfilling the role of actor A01 (the *completer* of the purchase of a pizza). But for practical reasons, the 'order taker' needs to delegate his authority for some of the C-acts he needs to perform: the 'deliverer' is delegated with the acceptance of T02 (the preparation transaction), the request and acceptance of T03 (the payment transaction), and the statement of T01 (the preparation transaction); the 'customer' is delegated with the acceptance of T04 (delivery transaction). Hence, the 'order taker', the 'deliverer' and the 'customer' work together to complete all the acts that A01 needs to perform. We could say that this situation is somehow alike to the fulfillment of actor roles, the collectively way. Compared with the approach from [Str93], the obligations in [Str93] are actually the C-acts from DEMO.

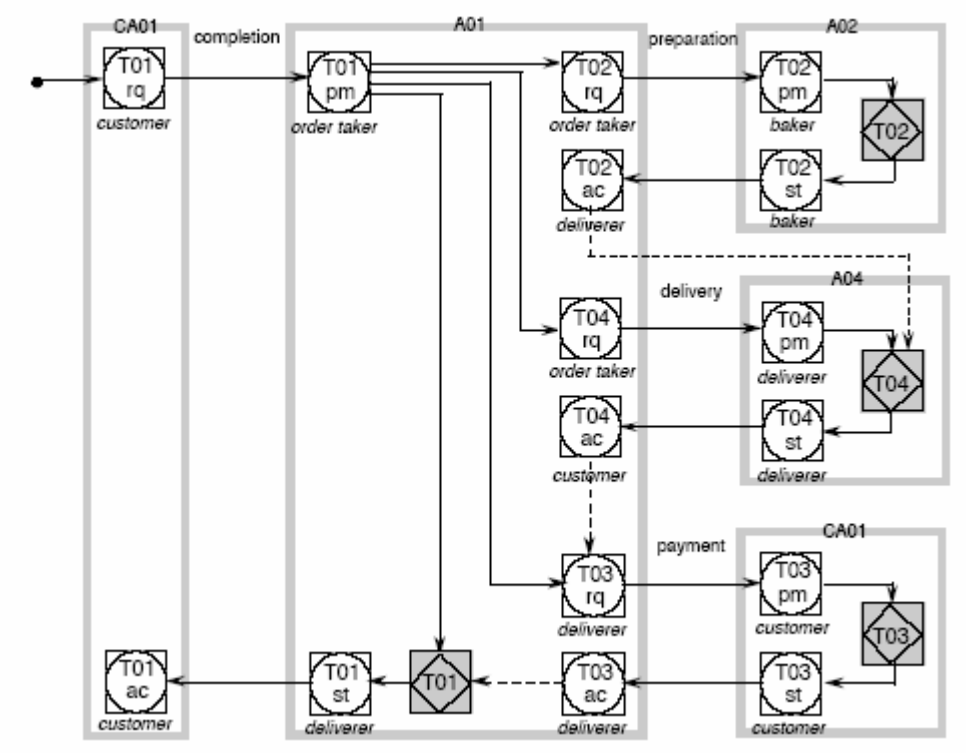


Figure 4-3: The process of delegation explained on the Pizzeria case study from DEMO [Die05a]

As previously mentioned, the subject that delegates his authority to perform some C-acts to another subject remains responsible for the correct completion of the transaction. This means, that the former subject is dependent on the latter. If we reason on the idea of the malicious actor, we can easily see how the point of delegation presents a vulnerability of the system. The 'order taker' bases its actions on the good-will of the 'deliverer' and the 'customer'. Even though he delegates the latter two to perform some acts, he is still responsible for the outcome. For example, if a malicious actor, the 'deliverer' can choose not to deliver the pizza to its destination. Therefore, to prevent these kinds of things to happen, an audit trail needs to be in place so that all the performed acts can be traced back to the 'guilty' actor.

Delegating authority also has to do with information sharing between the parties involved in the delegation process [Soh05]. As delegating the authority to perform a certain task means also transfer of information, the process should only be possible to be performed under certain conditions of authorization. Thus, certain constraints must be applied to the delegated person and only if it suffices those constraints, he can become the new authorized to perform certain facts.

4.5 Summary and discussions

In this chapter, we have attempted to extract and comment upon all the aspects related to the concept of responsibility in DEMO. We have first portrayed how DEMO defines its actors and actor roles and how the fulfillment of actor roles can generate into vulnerabilities of the system. We have seen how responsibility, authority and competence are defined and how the areas of responsibilities from the Process Model and the Interstriction Model can drive to the construction of well-defined and thorough security policies. We have seen how the process of delegation can trigger dependencies between the actors and how these dependencies constitute the vulnerable points of the system.

After having identified all these, the next natural question would be how can we learn from this analysis of DEMO and also from the results described in the previous chapter. How can the current research in responsibility modeling help us in establishing a way of modeling security starting from DEMO? To begin answering these questions, the following table, based on the same model as Table 3-1, will present and discuss a comparison between the modeling approaches described in Chapter 3 and DEMO in respect to how they refer to the issue of responsibility.

(1) The method	(2) Objectives	(3) Similarities with the DEMO approach to responsibility	(4) Differences from the DEMO approach to responsibility
Dobson's approach from 1993 [Str03] and the continuation : the later DIRC project	Defines the responsibilities and obligations within an organization, concentrating on building a 'need-to-know', 'need-to-do', 'need-to-audit' list for information. This way, the authors show the path to the foundation of security policies. It also identifies delegation as the process that triggers the vulnerable points of the system.	Both approaches are based on the speech act theory; [Str93] says that only obligations can be delegated and the one that delegates them is the responsible subject. In DEMO, authority to perform things can be delegated in the sense that the person authorized to act can delegate C-acts to someone else, but he/she will ultimately stay responsible. The idea of the both approaches is the same, just that the obligations in [Str93] are the acts in DEMO (a matter of notation).	DEMO integrates the study of responsibility in the larger context of designing the enterprise whereas [Str93] does not as it only concentrates on deriving security requirements from the analysis of responsibilities.
Thomas and Sandhu in 1994 [Tho94]	Concentrates on responsibilities and authorization structures to build a better authorization model from an enterprise perspective.	Both approaches tackle the issues of responsibility and authorization, although the former is much too technical oriented.	[Thos94] concentrates mainly on technical issues and the main design objective is intended towards technical systems whereas DEMO's target is the social system.

(1)	(2)	(3)	(4)
Backhouse and Dhillon in 1996 [Bac96]	Reasons for the idea that analyzing the structures of responsibility in an organization would lead to the development of secure information systems. The approach is an interpretative one and presents a "framework for security evaluation based on ontological interpretation".	Both approaches use responsibility when trying to model the organization.	The approach in [Bac96] concentrates on the ontological dependencies between the nodes and thus, the approach would be hard to introduce, for example, within the DEMO models or any other methodology. Contrary to that, DEMO integrates responsibility modeling within its models when designing the whole enterprise.
McDermott and Fox in 1999 [Mcd99]	Adapts use cases in abuse cases to analyze the security requirements in a simple way that is accessible for non security specialists	They both try to analyze the system they refer to in a way that is easy to understand for non experts.	Use cases are mainly used in software engineering whereas DEMO is a methodology for modeling organizations as social systems.
The i* approach [Liu03a]	Claims that due to the fact that security and privacy issues are a result of human "concerns and intents", they should be modeled through social concepts. To accomplish this, the authors propose a methodological framework for analyzing security and privacy based on the concept of strategic social actors.	Both approaches make use of the idea that "only humans can create new things".	Within the i* framework, there is not a clear separation between humans and other kinds of agents: both are called agents and they can be human or software agents. In DEMO, this separation is very clear and very important.

Table 4-1: Responsibility modeling approaches of security and the DEMO approach to responsibility

Given the results discussed in the table above, we can now reason on how they can help in establishing a starting point for modeling security within DEMO. The comparisons in the table take the line of analysis to the suggestion that there are in fact two approaches whose ideas can contribute to this work: [Str93] and [Liu03].

The approach in [Str93] uses the responsibilities and obligations within an organization to derive security requirements. As we have identified, the obligations in [Str93] are the equivalent of C-acts in DEMO. Therefore, from the responsibility areas as they are defined in the Process Model, where all the acts an actor role needs and is allowed to perform are described, the security policy for 'need-to-know' and 'need-to-do' can be established. This way, by having

properly defined security policies, the created system can resist to various attacks if those policies properly enforced. The issue of transfer of obligation (transfer of authority, so the delegation of C-acts) and fulfillment of actor roles generate the need of the 'need-to-audit' list. This way, we establish a security policy for audit trail that will provide a way of being able to identify all the acts and who performed them. This will help the system when it needs to recover from an attack: it offers the possibility of identifying who did what and who is accountable for the breach in the system. Hence, from direct responsibility analysis we have learnt that in order to derive a good security policy, we need start by identifying the actor roles with their responsibilities and their obligations (C-acts in DEMO). We have also learnt that the process of transfer of obligation (transfer of authority in DEMO) can affect the security of the system.

The approach in [Liu03] provides us with a general line of integrating a tactic of detecting the possible attacks on the system (by transforming the actors into attackers, by identifying the points of dependability between the actors).

Following this line of reasoning, an approach to security starting from the DEMO models would need to follow the next steps:

1. Identify the actor roles: DEMO makes a very good identification of all the actor roles;
2. Identify the areas of responsibility: who is allowed to do what, who is allowed to see what (which banks are allowed to be accessed and by whom), and which are the things that need to be recorded for audit; the Process Model and the Interstriction Model came here as important ;
3. From the areas of responsibility with all the elements included at step 2, the lists of 'need-to-know', 'need-to-do', and 'need-to-audit' can be identified as requirements for security;
4. Perform an attacker analysis: transform each actor role in an attacker and identify how it could hurt the system;
5. Identify the vulnerabilities of the studied system;
6. Identify threat scenarios and present how could system's failure be prevented.

The previously presented steps propose a possible approach for dealing with security within a DEMO context. Next, to demonstrate its feasibility, it will be applied on two case studies in the next chapters. First, we will detail the well-known library case from DEMO (Chapter 5) and second, we will use a real-life study case, the Import/Export Services Company (Chapter 6).

The case study is one of the ways of doing social science research [Yin94].
It is "an in-depth exploration of one particular case (situation or subject)
for the purpose of gaining depth of understanding into the issues being investigated" [etr]

5 Case study 1 – the Library

This Chapter will portray the first case study taken into consideration in this thesis' work. First, the reader will be provided with a description and explanation of the case within the DEMO context. The next step will consist in applying the security analysis as proposed in the previous Chapter. Lastly, the findings will be summarized and discussed.

5.1 Description and explanation within the DEMO context

The Library case study is one of the two examples that that spiritual father of DEMO, Professor Jan Dietz, uses in [Die05a] to explain how DEMO works. For the reader's convenience, if curious, Annex 1 offers a complete excerpt of the case together with the DEMO diagrams necessary for this study.

The case envisions how a university's library functions and operates. The actors involved are presented in Table 5-1. There are operations like loaning books, returning books, applying for a card, registering, paying for the card, paying a fine if necessary, etc. Hence, the transactions are the ones in Table 5-2.

Actor role notation	**Actor role description**
CA01	The board of the library
CA02	Aspirant member
CA03	Publisher
CA04	Member
A01	Registrar
A04	Loan creator
A06	Loan terminator
A09	Stock controller
A10	Annual fee controller

Table 5-1: The actor roles in the library case [Die05a]

Transaction type	**Resulting P-fact type**
T01 membership registration	PF01 membership M has been started
T02 membership fee payment	PF02 the fee for membership M in year Y has been paid
T03 reduced fee approval	PF03 the reduced fee for M in year Y is approved
T04 loan start	PF04 loan L has been started
T05 book return	PF05 book copy C has been returned
T06 loan end	PF06 loan L has been ended
T07 return fine payment	PF07 the late return fine fee for loan L has been paid
T08 book shipment	PF08 shipment S has been performed
T09 stock control	PF09 the stock control for month M has been done
T10 annual fee control	PF10 the annual fee control for year Y has been done

Table 5-2: The Transaction Result Table [Die05a]

5.2 Security analysis

Starting from the line of working proposed at the end of Chapter 4, we can see that, in fact, the first two points are already performed in DEMO. So, the first step that needs to be completed and which is in direct regard to security is **step 3**: the lists of 'need-to-know', 'need-to-do', and 'need-to-audit'. Taking the results from the Process Model (responsibility areas) and the Interstriction Model as they are depicted in [Die05a] and, for the reader's convenience, replicated in Annex 1, the following lists are obtained:

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
CA01 - The board of the library	From Annex 1 - 3: each time A01 ('registrar') requests a reduced fee approval, CA01 needs to deal with that fact - promise, execution, state of T03 From Annex 1 - 6: each time A10 ('annual fee controller') requests a reduced fee approval, CA01 needs to deal with that fact - promise, execution, state of T03	From Annex 1 - 7 and Annex 1 - 8: the elements in CB03 and PB03 - all the production facts and coordination facts that are stored in CB03 and PB03 at different times	Relative time + promise, execution and state of T03

Table 5-3: list of "need-to" for actor role 'The board of the library'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
CA02 - Aspirant member	From Annex 1 - 3: each time A01 ('registrar') requests membership fee payment, CA02 needs to deal with that fact - promise, execution, state of T02 From Annex 1 - 3: to become a member, CA02 needs to request T01 ('membership registration') and also needs to accept T01 if the case From Annex 1 - 6: each time A10 ('annual fee controller') requests a membership fee payment, CA02 needs to deal with that fact - promise, execution, state of T02	From Annex 1 - 7 and Annex 1 - 8: the elements in CB02 and PB02 - all the production facts and coordination facts that are stored in CB03 and PB03 at different times	Relative time + request and accept of T01, promise, execution and state of T02

Table 5-4: list of "need-to" for actor role 'Aspirant member'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
CA03 – Publisher	From Annex 1 - 5: each time A09 requests for a book shipment, CA03 needs to deal with that fact – promise, execution and state of T08	From Annex 1 - 7 and Annex 1 - 8: the elements in CB08 and PB08 - all the production facts and coordination facts that are stored in CB08 and PB08 at different times	Relative time + promise, execution and state of T08

Table 5-5: list of "need-to" for actor role 'Publisher'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
CA04 – Member	From Annex 1 - 4: each time A06 requests for a return fee payment, CA04 needs to deal with promise, execution and state of T07 From Annex 1 - 4: each time CA04 wants to return a book, it needs to deal with request and accept T06 From Annex 1 - 2: each time CA04 wants to borrow a book, it needs to deal with request and accept T04 From Annex 1 - 2: each time A04 requests for a book return, CA04 needs to deal with promise, execution and state of T05	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB04, CB05, CB06, CB07 and PB08, PB05, PB06, PB07 at different times	Relative time + request and accept of T06 and promise, execution and state of T07 Relative time + request and accept of T04 and promise, execution and state of T05

Table 5-6: list of "need-to" for actor role 'Member'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A01 – Registrar	From Annex 1 - 3: each time CA02 requests for a membership registration, A01 needs to promise T01, and request and accept T03 (if the case). It also needs to accept T02 and execute and state T01	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB03, CB01, CB02, CPB11, CPB12, CPB14, PB03, PB01, PB02 at different times	Relative time + promise , execution and state of T01, request and accept of T03 if the case, request and accept T02

Table 5-7: list of "need-to" for actor role 'Registrar'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A04 – Loan creator	From Annex 1 - 2: each time CA04 requests for a loan start, A04 needs to promise T04 and request T05. If the case, it will need to execute and state T04, and to accept T05	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB04, CB05, CB06, CB08, PB04, PB05, PB06, PB08, CPB12, CPB14	Relative time + promise, execution and state of T04, request and accept of T05 if the case

Table 5-8: list of "need-to" for actor role 'Loan creator'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A06 – Loan terminator	From Annex 1 - 4: each time CA04 requests for a loan end, 06 needs to promise T06, and if the case, request and accept T07 and execute and state T06	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB04, CB05, CB06, CB07, CB08, PB04, PB05, PB06, PB07, PB08, CPB12, CPB14	Relative time + promise, execution and state of T06 and if the case, request and accept of T07

Table 5-9: list of "need-to" for actor role 'Loan terminator'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A09 – Stock controller	From Annex 1 - 5: each year, A09 needs to perform stock control. Thus, it needs to request, promise, execute, state and accept T09. And also, if the case, to request and accept T08.	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB09, CB08, PB09, PB08, CPB13, CPB14	Relative time + request, promise, execution, state and accept of T09; and request and accept of T08 when the case.

Table 5-10: list of "need-to" for actor role 'Stock controller'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A10 - Annual fee controller	From Annex 1 - 6: each year, A10 needs to perform and annual fee control. Thus, it needs to request, promise, execute, state and accept T10. And also, when the case, it needs to request and accept T03, request and accept T02.	From Annex 1 - 7 and Annex 1 - 8: all the production facts and coordination facts that are stored in CB03, CB04, CB10, PB03, PB04, PB10, CPB11, CPB14	Relative time + request, promise, execution, state and accept of T10; request and accept of T03 when the case; and request and accept T02 when the case

Table 5-11: list of "need-to" for actor role 'Annual fee controller'

The above lists can be further transformed in eligible security rules, like authorization rules (who is allowed to access what information from the list of 'need-to-know'). The elements from the first two columns mean that those are the facts and information an actor role is allowed to perform and know, respectively. Consequently, if the need-to-know and need-to-do do not exist, they are no allowed to see and do anything else than what the table specifies. This way, DEMO enforces a role-based access control.

Step 4 proposed an analysis of the actors as attackers and the possibilities they have to harm the system. Each actor fulfilling an actor role can use its capabilities in a malicious manner. Table 5-12 presents how each actor role can pose a threat to the analyzed system. The idea is taken from the i* framework proposed in [Liu03] which bases its approach on the assumption that an actor is "guilty until proven innocent" and thus measures have to be taken in order for the system to be protected from the attacks they can create. There needs to be mentioned that an attacker is not only characterized by malicious intent: an actor is regarded as an attacker also if we consider the unintentional actions that could harm the system.

The results in Table 5-12 are based on the assumption that an actor role can only perform action within its own area of responsibility. This is because the security restrictions imposed by the 'need-to' lists, take care that an actor fulfilling and actor role cannot act outside its responsibility area based on authorization rules. Thus, they can only use their capabilities and authority to act in a malicious way within their own area of responsibility. In addition to this, there is also the fact that we cannot really control the acts of the actors external to the system (outside the boundary): their actions are enforced by the already specified authorization rules. Therefore, the table bellow will only consider the internal actor roles as possible attackers of the system.

The possibilities of an attacker are to a certain extent limited by the rules imposed by the 'need-to' lists. As a consequence, the way one can harm the system is in direct relation with the quality of information one actor role produces and how the wrong information can trigger a list of vulnerabilities to the system. This attacker analysis also spots the dependencies between the actor roles.

Attacker actor role	Description
(A01 Attacker) Registrar	- has access to CPB11 (*personal data*), CPB12 (*library data*) and CPB14 (*general data*) → can tamper with data: disclosure of data to third parties; Note: the information in CPB11 can be derived from PB01 (all the personal data of the aspirants members that have registered) → A01 can also tamper with data in the sense of introducing the wrong data - it is the executor of T01 and the initiator of T02 and T03 → the quality of the facts A01 produces can affect the operations of all the actors roles that base their actions on the production and coordination facts produced by A01, like CA01, CA02 and A10
(A04 Attacker) Loan Creator	- has access to CPB12 (*library data*) → can tamper with data: disclosure of data to third parties; - it is the executor T04 and the initiator of T05 → the quality of the facts A04 produces can affect the operations of CA04 and A04 - has access to PB08, PB06: the 'need-to' lists from step3 enforced a 'read' access → it can generate the case of disclosure of information to third parties
(A06 Attacker) Loan Terminator	- has access to CPB12 (*library data*) → can tamper with data: disclosure of data to third parties; - it is the executor of T06 and initiator of T07 → the quality of the facts A06 produces can affect the operations of CA04 and A06 - has access to PB08, PB04, PB05: the 'need-to' lists from step3 enforced a 'read' access → it can generate the case of disclosure of information to third parties
(A09 Attacker) Stock Controller	- has access to CPB13 (*book titles*) → tampering with data: disclosure of data; Note: the information in CPB11 can be derived from the PB08 → A09 can tamper with data in PB08 - it is the initiator of T08 and executor/initiator of T09 → the quality of the facts he produces can affect the operations of A06 and A04
(A10 Attacker) Annual Fee controller	- it is the initiator/executor of T10 and the initiator of T01 and T02 → the quality of the facts he produces can affect the operations of CA01 and CA02 - has access to PB03, PB02

Table 5-12: Attacker analysis

Note: conform with Annex 1 - 8, all the actor roles have authorized access to CPB14 (*general data*) → tampering with data in the sense of disclosure of information

Taking into account the CIA security triangle (Section 3.4), it can be easily deduced that the threats that can appear in the library case have to do with the breach of the integrity, confidentiality and availability of the data: in the production and coordination banks. As a first measure to predict a possible breach due to disclosure of information is to impose a security policy in the form of a signed contract with each actor fulfilling an actor role regarding the secrecy of information.

A vulnerability would be a "weakness of an asset of a group of assets that a threat may exploit". For example, the lack of authorization rules for the access of the coordination and production banks can cause a breach in the integrity of data of the library. The results of the responsibility analysis that generated the 'need-to' lists have taken care of some of the vulnerabilities. There are still left the ones that are caused due to the dependencies between the actor roles as attackers identified in the analysis performed in Table 5-12 (the dependencies with

regard to the fulfillment of actor roles are taken care for by the enforcement of an audit policy as results from Tables 5.1 - 5.11 but only to a certain extent). The idea is how to prevent that those situations when an attacker tampers with the system do not affect the entire system. Table 5-13 depicts the vulnerabilities that can emerge as a result of these kinds of problems (**step 5**).

Note: the vulnerabilities and their clarifications refer to the general situation. Their application for this case is only for some of the examples presented (for example, there is no situation of delegation in the library's case, but the issue represents a vulnerability of a system as we have explained also in Section 4.4).

Vulnerability	**Clarification**
Inaccurate information	In Table 5-12 we have seen how the wrong information (usually with malicious intent but not only) can affect the work of other actors. So, it should be considered a rule that before using information in doing their jobs, the users would need to be sure that the information is accurate and has not been altered (validation of information). The line of thinking to a solution that would solve such a problem takes us to the idea that each time o production or coordination fact gets to be written in one of the corresponding banks, there should be a third party involved to check, validate and guaranty the accuracy of the information. This idea has it roots in the theory related to how one can guarantee the validity of an authentication protocol[5], in this case how can one guarantee the validity of information.
Broken access control	It is very important that the authorization rules as they derive from the 'need-to' lists to be properly enforced. As also mentioned before, there is also the need for them to be enforced more than at a role level. If not properly enforced, an attacker could use its rights to penetrate the system and tamper with information. There should be constant checks performed to assure the right implementation of the 'need-to' lists.
Delegation process	The delegation process in DEMO is very sensitive as it supposes the transfer of authorization to perform certain C-acts. This means also the transfer of the authorization for access and modification of certain information. There should be special delegation rules enforced to assure that this transfer is properly implemented. We know that the one that delegates ultimately remains responsible for the good execution of one act. In order to know exactly which actor has executed the act, the 'need-to-audit' list needs to contain not only what an actor role is allowed to perform but also a track of which user (actor) has actually performed it and not only which actor role. DEMO argues that the process of delegation together with all the matters that have to do with the fulfillment of actor roles are implementation issues and are abstracted from the ontological model. Thus, we only want point out the importance of them, and leave their elucidation to possible future work.

[5] idea originating from discussions with Ing. Rob Weemhoff. Mr. Weemhoff is an active member of a working group of the Dutch member body of the ISO, NEN, the counterpart of ISO/IEC JTC1 SC27 security techniques.

Vulnerability	Clarification
Multiple authorization rights	It can be the case that the same actor needs to fulfill two or more actor roles. For example, John can be the registrar, the loan creator and loan terminator in the same time. This way, all the authorization rights assigned to the three roles are actually assigned to the same actor. As an attacker, John can now hurt the system from three different roles. This is important when the roles are conflicting from an information point of view: one role should not have access to the information the other has.

Table 5-13: Vulnerabilities

Step 6 proposes to identify possible scenarios when the vulnerable points of the system are attacked and how they can be protected. The following table envisions a few threat scenarios when the library is under attack and how the measures discussed beforehand will prevent its failure:

Scenario description	System behavior
John and Mary fulfill the role of the registrar in a sequential manner: John works from 7 a.m. till 1 p.m. and Mary from 1 p.m. till 7. p.m. Scenario a): John needs to register a new member (Paul), but mistakenly or maliciously he inputs some faulty data with regard to the fee Paul needs to pay (due to the data introduced, the 'Board' will not approve 'reduced fee' to Paul although he is entitled to one).	It can be the case that Paul knows he is entitled to a reduced fee -> he can draw John's attention and the situation is solved. If not, all the acts and subsequent facts involved in 'reduced fee approval' and 'membership fee payment' will be false as they are based on a faulty information. This is an 'inaccurate information' type of vulnerability that has been attacked by actor role registrar. 1. if anytime discovered the error (on whichever way), due to the audit trail, John can be hold responsible for the wrong information he imputed -> thus, due to the 'need-to' lists, the system is prepared to recover from such an attack 2. prevention of errors in such cases could be solved by only allowing the operator to introduce only certain data: a policy on how the access to data is allowed can be derived from here (i.e. if birth year included, only allow certain values – validation of parameters). 3. what if such an error is produced and not discovered? What if John chooses to introduce faulty data with regard to the fees anyone should pay for some (unimportant) reason? His 'mistake' would propagate further in the system, because the 'board' will use wrong information to carry on its job, thus the integrity of data is breached. The system has the possibility of recovering from such an attack (see situation 1) if the mistake discovered. To this kind of error, the solution would be regular update of data.

Scenario description	System behavior
Scenario b): Mary only works at this library as an attacker (whichever the reason): she tampers the data, as she has access to 'personal data' and to 'library data' as a registrar	Although mall intentioned, each act Mary performs is registered (audit trail). For example, she modifies the data for Paul so that his birth year is changed. The system registers each time Mary enters the system and the modifications she makes so if the case, her manipulation can be traced – audit trail. This kind of vulnerability has to do with inaccurate information and the solution to an attack on it would be the existence of a third which verifies the validity of the production act and the fact produced (future work idea): each time a fact is produced, there is a third party that verifies the validity of such an act so that the data integrity is maintained.
Scenario c): John register Paul's data half way, his turn finishes and Mary gets to fill in the rest of the data. Mary could choose not to finish John's job and therefore, Paul's data are not complete/correct (integrity of data)	Enforced by the 'need to do' list, the actor role registrar needs to register all Paul's data. In addition to this, by having a trace audit ('need to audit'), Paul's acts are registered: his last act would be the message of continuation of his work left to Mary. Mary is thus forced to continue the work and the integrity of data is preserved.

Table 5-14: Threat scenarios

5.3 Summary

In this Chapter, the theory discussed in Chapter 4 has been applied on the Library case. We have seen this way how security requirements can be constructed starting from the DEMO models. Based on the diagrams in Annex 1, Tables 5.3 to 5.11 represent the starting points in driving security requirements: the authorization rules for information and the audit trail.

As we have explained before (Chapter 4), DEMO has already incorporated several points that drive to security specifications. From the Process Model and the Interstriction Model (Annex 1), one can easily derive security requirements for information, for how the authorization rules must be constructed and the establish basis for audit trail. Additionally, we only needed to reorganize the information in order to see how these rules apply to each actor role (Tables 5.3 to 5.11). Further on, we applied the method proposed at the end of Chapter 4 and we have identified where the system is vulnerable and discussed a couple of threat scenarios in order to see how the system can resist to attacks.

After discussing the work of this thesis on the Library case, next step is to make the same analysis in a real life case study, namely the Import/Export Service Company. This will make the subject of the next Chapter.

6 Case study 2 – the Import/Export Services Company

This Chapter will portray a second case study for the proposed method in the Chapter 4. First, the reader will be provided with a description and then the case will be explained and detailed within the DEMO context. The next step will consist in applying the security analysis as proposed in Chapter 4. Lastly, the findings will be summarized and discussed.

6.1 Description and explanation within the DEMO context

The Import /Export Services Company (IES) case study makes the object of the work performed in [Gal06]. Annex 2 offers all the DEMO diagrams necessary for the study of security on this case.

IES is a company situated in Baile Átha Luin with specialization in logistics. Its responsibility comprises the entire production delivery chain for its clients. We refer the reader to Annex 2 for a more detailed description of the company's activity and we further present the actors involved in the process and the transaction list as they were described in [Gal06].

Actor role notation	Actor role description
S1	Client
A01	Order deliverer (IES)
A02	Consignment deliverer
A03	Consignment packer
A04	Ship loader
A05	Shipping company
A06	Stevedore
A07	Conveyance service
A08	Customs office
A09	Conveyor
A10	Transport service (IES)
A11	Consignment controller
A12	Shipment controller
A13	Conveyance controller
A14	Loading controller

Table 6-1: The actor roles in the IES case [Gal06]

Transaction type	Resulting P-fact type
T01 Delivering client's order	F01 <ORD_CL> is delivered to the client
T02 Delivering consignment	F02 <ORD_IES> is delivered to the IES
T03 Packing consignment into container(s)	F03 <ORD_IES> is loaded into container(s)
T04 Loading container(s)	F04 <CNTR> is loaded onto ship
T05 Sailing of the ship	F05 <SH> has been sent to Luimneach
T06 Unloading the ship	F06 <SH> is unloaded
T07 Delivering container(s) to the IES	F07 <CNTR> is delivered to the IES
T08 Customs clearing of container(s)	F08 <CNTR> is customs cleared
T09 Transporting container(s) to the IES	F09 <CNTR> is transported to the IES
T10 Transporting client's order to the client	F10 <ORD_CL> is transported to the client
T11 Controlling consignments	F11 All the consignments for period <P_CONS> are checked
T12 Controlling shipments	F12 All the shipments for period <P_SHIP> are checked
T13 Controlling transportations	F13 All the transportations for period <P_TRAN> are checked
T14 Controlling loadings	F14 All the loadings for period <P_LOAD> are checked

Table 6-2: The Transaction Result Table [Gal06]

6.2 Security analysis

Following the same line of analysis as in the Library's case, we begin with step 3: the 'need-to' lists. Thus, according to the DEMO diagrams in the Process Model and Interstriction Model, the following lists are obtained:

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
S01 – Client	From Annex 2 - 3: each time A01 ('order deliverer IES') states it has delivered an order to the client, S1 needs to deal with accept T01 ('delivering client's order'); and each time S1 needs for an order to be delivered, it needs to request T01	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB01 and PB01	Relative time + accept of T01, request of T01

Table 6-3: list of "need-to" for actor role 'Client'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A01 – Order deliverer	From Annex 2 - 3: each time S1 ('client') requests the delivering of an order, A01 ('order deliverer IES') needs to promise T01 (if the case: before that needs to check whether T02 ('delivering consignment') has been accepted) and to request T10 ('transporting client's order to the client') to A10 ('transport service IES'). If the case, it will need to deal with accept T10, and execute and state T01.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB01, CB1, CB10, PB01, PB10	Relative time + promise, execute and state T01; request and accept T10

Table 6-4: list of "need-to" for actor role 'Order deliverer'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A02 – Consignment deliverer	From Annex 2 - 3: each time A11 ('consignment controller') requests T02 ('delivering consignment'), A02 needs to promise T02 and request T03 ('packing consignment into container'). Then it will need to deal with accept T03 and, if the case, execute and state T02.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB02, CB03, CB07, PB02, PB03, PB07	Relative time + promise, execute and state T02; request and accept T03

Table 6-5: list of "need-to" for actor role 'Consignment deliverer'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A03 - Consignment packer	From Annex 2 - 3: each time A02 requests T03 ('packing consignment into container'), A03 needs to deal with promise, execute and state T03	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB03 and PB03	Relative time + promise, execute and state of T03

Table 6-6: list of "need-to" for actor role 'Consignment packer'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A04 – Ship loader	From Annex 2 - 3: each time A14 ('loading controller') requests T04 ('loading container'), A04 needs to deal with promise, execute and state T04.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB04, CB4, PB04	Relative time + promise, execute and state of T04

Table 6-7: list of "need-to" for actor role 'Ship loader'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A05 – Shipping company	From Annex 2 - 3: each time A12 ('shipment controller') requests T05 ('sailing of the ship'), A05 needs to deal with promise, execute and state T05.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB05, PB05, CB10_	Relative time + promise, execute and state of T05

Table 6-8: list of "need-to" for actor role 'Shipping company'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A06 – Stevedore	From Annex 2 - 3: each time A12 ('shipment controller') requests T06 ('unloading the ship'), A06 needs to deal with promise, execute and state of T06.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB11_ CB06, PB06	Relative time + promise, execute and state of T06

Table 6-9: list of "need-to" for actor role 'Stevedore'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A07 – Conveyance service	From Annex 2 - 3: each time A13 ('conveyance controller') requests T07 ('delivering containers to the IES'), A07 will need to promise T07 and deal with request and accept T08, and request and accept T09 (if the case). Then, if T09 accepted, it will need to execute and state T07.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB07, CB08, CB09, CB7, PB07, PB08, PB09	Relative time + promise, execute and state of T07; request and accept of T08; request and accept of T09

Table 6-10: list of "need-to" for actor role 'Conveyance service'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A08 - Customs office	From Annex 2 - 3: each time A07 ('conveyance service') requests T08 ('customs clearing of container'), A08 needs to deal with that fact: promise, execute and state T08	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB08, CB5, PB08	Relative time + promise, execute and state of T08

Table 6-11: list of "need-to" for actor role 'Customs office'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A09 – Conveyor	From Annex 2 - 3: each time A07 ('conveyance service') requests T09 ('transporting container to IES'), A09 needs to promise T09 and then, if T08 was accepted, will need to execute and state T09.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB09, CB6, PB09	Relative time + promise, execute and state of T09

Table 6-12: list of "need-to" for actor role 'Conveyor'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A10 – Transport service (IES)	From Annex 2 - 3: each time A01 ('order deliverer IES') requests T10 ('transporting client's order to the client'), A10 needs to deal with promise, execute and state T10.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB10, CB07, CB02, CB2, PB10, PB07, PB02	Relative time + promise, execute and state of T10

Table 6-13: list of "need-to" for actor role 'Transport service (IES)'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A11 - Consignment controller	From Annex 2 - 3: periodically, A11 needs to request T11 ('controlling consignments'). Thus, each time A11 requests T11, A11 needs to deal with promise T11 and request and accept T02 ('delivering consignment'). Then, if T02 accepted, it needs to deal with execute, state and accept T11.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB01, CB11, CB3, PB01, PB11	Relative time + request, promise, execute, state and accept of T11; request and accept of T02

Table 6-14: list of "need-to" for actor role 'Consignment controller'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A12 – Shipment controller	From Annex 2 - 3: periodically, A12 needs to request T12 ('controlling shipments'). Thus, each time A12 requests T12, A12 needs deal with promise T12, request and accept T05 ('sailing of the ship') and, if T05 accepted, with request and accept T06 ('unloading the ship'). If T06 accepted, it will need to execute and state T12.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB12, CB04, CB05, CB06, CB9, CB10, PB12, PB04, PB05, PB06, PB10	Relative time + request, promise, execute, state and accept of T12; request and accept T05; request and accept T06

Table 6-15: list of "need-to" for actor role 'Shipment controller'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A13 – Conveyance controller	From Annex 2 - 3: periodically, A13 needs to request T13 ('controlling transportations'). Thus, each time A13 requests T13, A13 needs to deal with promise T13, and request and accept T07 ('delivering containers to IES'). If T07 accepted, A13 will need to deal with execute, state and accept T13.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB07, CB13, CB06, CB8, PB07, PB13, PB06	Relative time + request, promise, execute, state and accept of T13; request and accept of T07

Table 6-16: list of "need-to" for actor role 'Conveyance controller'

Actor role	'need-to-do'	'need-to-know'	'need-to-audit'
A14 – Loading controller	From Annex 2 - 3: periodically, A14 needs to request T14 ('controlling loadings'). Thus, each time A14 requests T14, A14 needs to deal with promise T14, and request and accept T04 ('loading containers'). If T04 accepter, A14 will need to deal with execute, state and accept T14.	From Annex 2 - 4 and Annex 2 - 5: all coordination and production facts in CB14, CB04, CB4, PB14, PB04	Relative time + request, promise, execute, state and accept of T14; request and accept T04

Table 6-17: list of "need-to" for actor role 'Loading controller'

The above lists can be further transformed in eligible security rules, like authorization rules (who is allowed to access what information from the list of 'need-to-know'). The elements from the first two columns mean that those are the facts and information an actor role is allowed to perform and know, respectively. Consequently, if the need-to-know and need-to-do do not exist, they are no allowed to see and do anything else than what the table specifies. This way, DEMO enforces a role-based access control.

Step 4 proposed an analysis of the actors as attackers and the possibilities they have to hurt the system. Each actor fulfilling an actor role can use its capabilities in a malicious manner. In the same manner we have done in the library case, Table 6-18 presents how each actor role can pose a threat to the analyzed system.

Attacker actor role	Description
(A01 Attacker) Order deliverer IES	- has access to CB1 (*IES clients' database*) – can tamper with data; - it is the executor of T01 and the initiator of T10 → the quality of the facts A01 produces can affect the operations of all the actors roles that base their actions on the production and coordination facts produced by A01, like A10 and A11; - has access to PB10: the 'need-to' lists from step3 enforced a 'read' access for A01 to access PB10;
(A02 Attacker) Consignment deliverer	- it is the executor T02 and the initiator of T03 → the quality of the facts A02 produces can affect the operations of A03, A04, A10, A11; - has access to PB07, PB03: the 'need-to' lists from step3 enforced a 'read' access for A02 to access the two banks;

Attacker actor role	Description
(A03 Attacker) Consignment packer	- it is the executor of T03 → the quality of the facts A06 produces can affect the operations of A02 and A04;
(A04 Attacker) Ship loader	- it is the executor of T04 → the quality of the facts he produces can affect the operations of A14 and A12;
(A05 Attacker) Shipping company	- has access to CB10_ (*Ship register*) – can tamper with data; - it is the executor of T05 → the quality of the facts he produces can affect the operations of A12;
(A06 Attacker) Stevedore	- has access to CB11_ (*Port storage facilities*) – can tamper with data; - it is the executor of T06 → the quality of the facts he produces can affect the operations of A12 and A13;
(A07 Attacker) Conveyance service	- has access to CB7 (*IES storage facilities*) – can tamper with data; - it is the initiator of T08 and T09 and the executor of T07→ the quality of the facts he produces can affect the operations of A08, A09, A13, A10 and A02; - has access to PB08, PB09: the 'need-to' lists from step3 enforced a 'read' access for A07 to access the two banks;
(A08 Attacker) Customs office	- has access to CB5 (*Customs regulations*) – can tamper with data; - it is the executor of T08 → the quality of the facts he produces can affect the operations of A07;
(A09 Attacker) Conveyor	- has access to CB6 (*Conveyor's transport means park*) – can tamper with data; - it is the executor of T09 → the quality of the facts he produces can affect the operations of A07;
(A10 Attacker) Transport office IES	- has access to CB2 (*IES transport means park*) – can tamper with data; - it is the executor of T10 → the quality of the facts he produces can affect the operations of A01; - has access to PB01, PB02, PB07: the 'need-to' lists from step3 enforced a 'read' access for A07 to access the three banks;
(A11 Attacker) Consignment controller	- has access to CB3 (*Consignment list*) – can tamper with data; - it is the initiator/executor of T11, and the initiator of T02 → the quality of the facts he produces can affect the operations of A02 and A10;
(A12 Attacker) Shipment controller	- has access to CB9 (*Shipments list*) – can tamper with data; - it is the initiator/executor of T12, and the initiator of T05 and T06 → the quality of the facts he produces can affect the operations of A05, A06, A13;
(A13 Attacker) Conveyance controller	- has access to CB8 (*Conveyance list*) – can tamper with data; - it is the initiator/executor of T13, and the executor of T07 → the quality of the facts he produces can affect the operations of A02 and A10;

Attacker actor role	Description
(A14 Attacker) Loading controller	- has access to CB4 (*Loadings list*) – can tamper with data; - it is the initiator/executor of T14, and the initiator of T04 → the quality of the facts he produces can affect the operations of A04 and A12;

Table 6-18: Attacker analysis

As we have reasoned in the Library' case, Table 5-13 depicts the vulnerabilities that can emerge in the general case (**step 5**). Next, **step 6** proposes to identify possible scenarios when the vulnerable points of the system are attacked and how they can be protected. The following table envisions two threat scenarios when the IES is under attack:

Scenario description	System behavior
Scenario a): John, the client, orders a DVD player. IES makes the order to the company in the Far East. The supplier puts together the consignment for that day, and Mary, representing the consignment packer, packs the products into containers, but she 'forgets' to pack John's DVD.	The 'need-to' lists forces Mary to pack John's DVD and this fact to be registered. But, as she plays here the attacker's role, she does not pack it. A02 cannot execute the delivery unless it accepts that John's DVD has been packed. In addition to this, A11, *consignment controller*, will periodically check that the consignments to ensure the proper processing of the goods. This way, if anything goes wrong it will be discovered on the way, and the audit trail will help the system recover from attack by identifying exactly where things went wrong and pointing out the 'guilty' person.
Scenario b): Mary, representing stevedore, when the ship arrives to Luimneach effectuates the unloading, but she effectuates the entire unloading except one. The conveyance follows its path to the IES without anyone to notice one container is missing.	This scenario has its countermeasure by having the *shipment controller* actor role effectuating a periodically check on the *shipping company*'s activities and *stevedore's* activities. The information is traced due to the enforced 'need-to-audit' list and again, the 'guilty' actor identified.

Table 6-19: Threat scenarios

6.3 Summary

In this Chapter, the theory discussed in Chapter 4 has been applied on a real life case – the IES. As in the Library's example, we have seen this way how security requirements (the 'need-to' lists) can be constructed starting from the DEMO models. Thus, from the Process Model and the Interstriction Model (Annex 2), one can easily derive security requirements for information, for how the authorization rules must be constructed and the establish basis for audit trail. As well, we only needed to reorganize the information in order to see how these rules apply to each actor role (Tables 6.3 to 5.17). Further on, we applied the method proposed at the end of Chapter 4 and discussed a couple of threat scenarios in order to see how the system can resist to attacks.

"Not everything that can be counted counts,
and not everything that counts can be counted"
Albert Einstein

7 Discussions

This thesis was aimed at exploring how security aspects within organizations can be addressed at a very high level: an ontological level that encapsulates construction and operation issues of organizations with no reference to implementation concerns. To do this, DEMO (Dynamic Engineering and Modeling for Organizations) has been found as the relevant methodology to use. Therefore, the main concern was to find and investigate the relationship between security and DEMO and to establish the first steps for modeling security concerns within a DEMO environment. The process was divided into 5 main research questions as presented in Table 7-1.

Research questions:	***Outcomes:***
1. To what extent does the current research in the field of security address the information systems security concerns?	• In the current research, information security concerns are solved primarily through technical solutions [Sip05b], bottom-up approaches: the research in information systems security still resides on the idea of the "organization as a machine" [Dhi01b]; • a new generation of information security methods is identified as necessary; • responsibility – "a key issue for security" [Str93]; • security approaches within the interpretive paradigm are identified as important for further analysis and discussion;
2. What is the connection thread between DEMO and Security?	Responsibility is identified as the thread that makes the connection between security and DEMO.
3. To what extent does the current research addresses the issue of 'responsibility – a security issue'?	Five main responsibility modeling techniques were identified in the literature and studied with their differences, their strengths and weaknesses. Some refer to technical systems and some do not include security modeling within the whole enterprise engineering process. The lessons learnt from these studies constitute the starting point for the fifth research question.
4. Which are the security aspects incorporated in the DEMO methodology?	• actors and actor roles; • the trio responsibility – authority – competence; • the Process Model and the areas of responsibility; • the process of delegation; • the Interstriction Model;
5. How could security aspects be tackled using the DEMO methodology?	The studied responsibility modeling approaches to security are combined into a step oriented method to tackle security at a DEMO level. This method is applied on DEMO's library example and on the IES example.

Table 7-1: Research questions and respective results

7.1 Findings

Research Question 1:
Question: To what extent does the current research in the field of security address and solve the information systems security concerns?
Findings:
The analyzed and studied literature revealed the fact that although research in information systems security has moved towards a socio-organizational perspective, **the approaches for managing security still reside on the idea of the "organization as a machine"** [Dhi01b]. To this end, the author in [Sip05b] brings together various literature examples in his attempt to say that security problems may rise from the "lack of fit" between social and technical approaches: security "is not just a technology problem, but it also involves people". His reasoning is based on a four generation classification of security methods: two first generations of traditional methods that "entail the technical view of the organizational role of information systems security" whereas the social-organizational nature is not seriously considered; a third generation of methods for modeling organizations' information systems security requirements (i.e. logical modeling, spiral approach, planning methodologies, the responsibility modeling, the task-based authorization, the abuse case); and the fourth generation including methods that add the socio-technical design aspect to the ones in the third generation (i.e. the user participation utilized in the soft approach to the planning of information systems security, semantic responsibility analysis, survivable information systems approach). The logic following this generations' split is aimed towards the idea of **a next generation of "social and adaptable information systems security methods".**

Taking a different approach but in the same line of ideas, the authors in [Dhi01b] argue that in order for security approaches to work in a "systematic and appropriate manner", it is essential to understand their "conceptual basis". To this end, the interpretive paradigm comes as interesting: it studies "the world as it is" - the social reality is "a network of assumptions and inter-subjectively shared meanings" and reality is a result of individuals' actions. Research directions within this paradigm include works that are based on the speech act theory (as portrayed in [Sea69]) to specify organizational security requirements (i.e. [Dob91], [Str93]). As DEMO also incorporates the speech act theory, we have considered the methods within this direction as base for further study. Within this line of work, **responsibility** is considered as **"a key issue for security"** [Str93]. Therefore, **security approaches within the interpretive paradigm are identified as important for further analysis and discussion**.

Research Question 2:
Question: Which is the connection thread between DEMO and Security?
Findings:
The second Chapter of this thesis has introduced the theory about DEMO and enterprise ontology. This way, we have seen that DEMO also encloses and explains the aspects of responsibility within an enterprise: organizations are social systems having as active elements or subjects the social individuals who behave "according to assigned authority and corresponding responsibility against a common background of social norms and values" [Die05]. Hence, as an answer to the second research question of this thesis, it has been stated that **responsibility is the thread connecting security with DEMO**. Further on, the report will consider responsibility analysis approaches to security when trying to address security at a DEMO level.

Research Question 3:
Question: To what extent does the current research addresses the issue of 'responsibility – a security issue'?
Findings:
There are mainly three responsibility modeling approaches of information systems security identified in the literature [Dhi01a]: (1) Dobson's approach from 1993 [Str03], (2) Thomas and Sandhu in 1994 [Tho94], and (3) Backhouse and Dhillon in 1996 [Bac96]. To these three, there are two more that, although they do not start directly from responsibility, can be integrated in the same line of thinking. These are (4) McDermott and Fox in 1999 [Mcd99] and (5) i* approach from 2003 [Liu03a]. The idea proposed by the last two starts from identifying the actors of the system in terms of their capabilities, which used in a malicious manner, can breach the security of the system. We have taken all these approaches and tried to constructively identify and discuss their relations with this thesis' objective, their differences, their strengths and their weaknesses.

The analysis concluded that all **the presented approaches to security start from the idea of responsibility** (the last two do not directly specify responsibility, but the actors include responsibilities – the approaches are similar to the others due to this characteristic). **Their shortcomings reside in the fact that they either refer to technical systems (and not social ones) or they do not include security modeling within the whole enterprise engineering process.** The lessons learnt from these studies generated the need of a critical analysis of DEMO in terms of responsibility in order to be able to establish the starting point for a possible method for analyzing security starting from DEMO (question 4).

Research Question 4:
Question: What are the security aspects incorporated in the DEMO methodology?
Findings:
As the answer of the second research question, responsibility has been identified as the thread connecting DEMO with security. Further on, we have identified and discussed all the aspects related to responsibility encapsulated in DEMO. This analysis revealed that DEMO incorporates aspects that are discussed in the literature as important to security: **actors, roles** and responsibility; the trio **responsibility – authority – competence**; the **Process Model** that defines the **areas** and structures **of responsibility** and also the **process of delegation** (considered to be a vulnerability of the system) is tackled; the **Interstriction Model** defines who is allowed to see what information.

These findings provided the basis for a comparison between DEMO's approach to responsibility and the previous analyzed security modeling approaches based on responsibility. We have taken the results of research questions 3 and 4 and formed the entries for providing an answer to the last research question.

Research question 5:
Question: How could security aspects be tackled using the DEMO methodology?
Findings:
We have begun answering this question based on the results in Table 4-1. The comparisons in the table took the line of analysis to the suggestion that there are in fact two approaches whose ideas can contribute to this work: [Str93] and [Liu03].

The approach in [Str93] uses the responsibilities and obligations within an organization to derive security requirements. As we have identified, the obligations in [Str93] are the equivalent of acts in DEMO. Therefore, from the responsibility areas as they are defined in the Process Model, where all the acts an actor role needs and is allowed to perform are described, the security policy for 'need-to-know' and 'need-to-do' can be established. This way, by having

properly defined security policies, the created system can resist to various attacks if those policies properly enforced. The issue of transfer of obligation (transfer of authority, so the delegation of C-acts) and fulfillment of actor roles generate the need of the 'need-to-audit' list. This way, we have established a security policy for audit trail that will provide a way of being able to identify all the acts and who performed them. This will help the system when it needs to recover from an attack: it offers the possibility of identifying who did what and who is accountable for the breach in the system. Hence, from direct responsibility analysis we have learnt that in order to derive a good security policy, we need start by identifying the actor roles with their responsibilities and their obligations (C-acts in DEMO). We have also learnt that the process of transfer of obligation (authority in DEMO) can affect the security of the system.

The approach in [Liu03] provides us with a general line of integrating a tactic of detecting the possible attacks on the system (by transforming the actors into attackers, by identifying the vulnerabilities in the system, the points of dependability between the actors).

Following this line of reasoning, we have identified **an approach to security starting from the DEMO models** that would need to follow the next steps:

1. Identify the actor roles: DEMO makes a very good identification of all the actor roles;
2. Identify the areas of responsibility: who is allowed to do what, who is allowed to see what (which banks are allowed to be accessed and by whom), and which are the things that need to be recorded for audit; the Process Model and the Interstriction Model came here as important ;
3. From the areas of responsibility with all the elements included at step 2, the lists of 'need-to-know', 'need-to-do', and 'need-to-audit' can be identified as requirements for security;
4. Perform an attacker analysis: transform each actor role in an attacker and identify how it could hurt the system;
5. Identify the vulnerabilities of the studied system;
6. Identify threat scenarios and present how could system's failure be prevented.

We have applied this method on two case studies: the well-known library case from DEMO (Chapter 5) and the real-life study case, the Import/Export Services Company (Chapter 6). Based on the analysis of these two cases, we have seen how the above detailed method works in practice and we have added an argument to sustain the opinion that this method can constitute a starting point when trying to tackle security at a DEMO level.

7.2 Limitations

As any research work, this thesis falls under the saying that "not everything that can be counted counts, and not everything that counts can be counted" [Albert Einstein]. It might be the case that some security information relevant to this approach has been left out and that some interpretations were subjective to our knowledge and literature study. We have tried, by the best of our capabilities, to overcome this by careful study of the literature in case and lucrative discussions with experts on the subject.

The main contributions of this thesis included finding the connection between DEMO and security, producing a critical analysis of DEMO from a security perspective and establishing a starting point for modeling security concerns within a DEMO environment. To this end, the method proposed at the end of Chapter 4 is not a very rigorous one: it represents mainly the line of thinking in such a context. It is considered and proposed as a matter of future study to take this line above its conceptual state and develop it into a thorough and systematic framework of addressing security at a DEMO level.

As a last point in this limitation Chapter, it might be argued that this thesis has crossed the ontology line by presenting threats scenarios that are based on implementation issues as they are defined in [Die05a]: i.e. the fulfillment of actor roles are defined as an implementation

issue in DEMO. This is why, as we have also explained in Chapter 4, we have thought that this line of implementation of actor roles are important in terms of expressing security concerns.

7.3 Main implications for further work

As also mentioned in the previous Section, the method defined in this thesis is not a very rigorously defined one: further work can imply the definition of a formal framework to deal with security within a DEMO context staring from the conceptual idea discussed at the end of Chapter 4. As practice often shows to be the one that defines the feasibility of an approach, we propose that this extended method to be further applied and proven on various case studies from different industry branches.

In Chapter 5 we have talked about the idea of the existence of a third party in the coordination act: a third party to check, validate and guaranty the accuracy of the information (idea that finds its routes in the theory related to how one can guarantee the validity of an authentication protocol). We think that further study and investigations in this direction would improve and extend the work performed in this thesis.

8 Conclusions

This thesis was aimed at exploring how security aspects within organizations can be addressed at a very high level: an ontological level that encapsulates construction and operation issues of organizations with no reference to implementation concerns. To do this, DEMO (Dynamic Engineering and Modeling for Organizations) has been found as the relevant methodology to use. Therefore, the main concern was to find the relationships between security and DEMO, to perform a critical analysis of DEMO from a security perspective and to establish a starting point for modeling security concerns within a DEMO environment.

The thesis is based on 5 research steps. We have started with a thorough study of information systems security issues. This research has identified the socio-organizational perspective as the future way of approaching security of information systems and has brought forward the fact that although research in information systems security has moved towards a socio-organizational perspective, the approaches for managing security still reside on the idea of the "organization as a machine" [Dhi01b]. This perspective is part of the interpretive paradigm which includes research directions within the speech act theory based on security modeling starting from responsibility. This way, the connection with DEMO was identified and we continued our research following this line.

We have then introduced the reader into DEMO's world by explaining concepts as enterprise ontology, the connection between security and ontology, the theory behind DEMO and the way of working in DEMO. We have this way managed to establish the thread that connects DEMO with Security as being responsibility. Therefore, further on we have studied and analyzed various approaches to model security starting from responsibility. After having discussed and compared five relevant methods, the next step was to perform a critical analysis of DEMO from a security perspective. This investigation revealed the fact that DEMO incorporates various aspects discussed in the literature as important to security, like actors, roles and responsibility; the trio responsibility – authority – competence; the Process Model that defines the areas and structures of responsibility; the process of delegation; the Interstriction Model. We have used these findings and compared DEMO's approach to responsibility with the previous analyzed security modeling approaches based on responsibility. The results of this comparison have constituted the entry arguments for establishing a starting point for modeling security within DEMO. Two case studies were used for illustrating purposes of the proposed method.

Although further research on the method suggested and more practical experiences are needed, the experience gained from the analysis performed in this thesis proved to be an important starting point when one tries to study security at a DEMO level.

References

[And93] Anderson, R., *Why cryptosystems fail*, Communication of the ACM, 37(11), 32–44, 1993

[Bac96] Backhouse, J., Dhillon, G., *Structures of responsibility and security of information systems*, European Journal of Information Systems, 1996

[Bar98] Barnes, B., *Computer security research: a British perspective*, IEEE Software 15(5), 30–33, 1998

[Bas03] Bass, L., Clements, P., Kazman, R., *Software architecture in practice*, Addison-Wesley Pearson Education, Second edition, 2003

[Bas91] Baskerville, R., *Risk analysis: an interpretative feasibility tool in justifying information systems security*. European Journal of Information Systems, 1991

[Bas93] Baskerville, R., *Information systems security design methods: implications for information systems development ,* ACM Press New York, 1993

[Bro79] Browne, P. *Security: Checklist for Computer Center Self Audits,* from Practicing software engineering in the 21st century, AFIPS Press, Arlington, VA., 1979

[Bur79] Burrell, G., Morgan, G., *Sociological Paradigms and Organizational Analysis*, Heinemann, 1979

[Cal05] Calder, A., Watkins, S., *IT Governance*, London and Sterling, VA, 3rd edition, 2005

[CC05] Common Criteria – Common criteria for Information Technology Security Evaluation, Version 2.3, August 2005

[Chi03] Chinburg, S.J., Sharda, R., Weiser, M., *Establishing the Business Values of Network Security Using Analytical Hierarchy Process,* Creating Business Value with Information Technology – Challenges and Solutions, Idea Group Publishing, 2003

[CISSP] The Certified Information Systems Security Professional (CISSP) certification of the International Information Systems Security Certification Consortium, Inc. ((ISC)2)

[Cla87] Clark, D., Wilson, D., *A comparison of Commercial and Military Computer Security Policies,* Proceedings of Symposium on Security and Privacy, IEEE Computer Society, 1987

[demo] http://www.demo.nl

[Dep04] UK Department of Trade and Industry, *Information Security Braches Survey*, 2004

[Dev00] Devambu, P., Stubblebine, S., *Software engineering for security. A roadmap.* International Conference on Software Engineering, ICSE, 2000

[Dhi01a] Dhillon, G., *Information Security Management: Global Challenges in the New Millennium,* IDEA Group Publishing, 2001

[Dhi01b] Dhillon, G., Backhouse, J., *Current directions in IS security research: towards socio-organizational perspectives*, Information Systems Journal, 2001

[Die00] Dietz, J.L.G., *Transaction Based Enterprise Modeling*, Proceedings of IFIP WCC 2000, 2000

[Die01] Dietz, J. L. G., *DEMO: towards a discipline of organisation engineering,* European Journal of Operational Research 128(2): 351-63, 2001

[Die03] Dietz, J. L. G., *Designing Technical Systems as Social Systems*, Proceedings of the 8th International Working Conference on the Language-Action Perspective on Communication Modelling (LAP 2003), July 1-2 2003, Tilburg, the Netherlands

[Die04] Dietz, J.L.G, Habing, N., *A meta ontology for organizations*, International Workshop on Modeling Inter-Organizational Systems (MIOS), Cyprus, 2004

[Die05a] Dietz, J.L.G., *Enterprise Ontology – theory and methodology*, 2005

[Die05b] Dietz, J.L.G., *The third wave*, Landelijk Architectuur Congres, NBC Nieuwegein, 2005

[Die05c] Dietz, Jan L.G. , *in4153 Architectural Design of Business Systems- lecture notes*, 2005

[Die96] Dietz, J.L.G., *Een reis door Kabouterland*, Alphen aan den Rijn : Samsom bedrijfsInformatie, 1996

[Die99] Dietz, J.L.G., *Understanding and Modeling Business Processes with DEMO.*, Proceedings of the 18th International Conference on Conceptual Modeling, 1999

[Die99a] Dietz, J.L.G., *DEMO Modelling Handbook*, 1999

[dirc] http://www.dirc.co.uk

[Dob05] Dobson, J., *Roles are responsibility relationships really*, IEEE Symposium on People and Computers, October 2005

[Dob91] Dobson, J., *A methodology for analyzing human and computer related issues in secure systems,* Computer Security and Information Integrity by Dittrich, K.,, Elsevier Science Publishers, 1991

[Don03] Donner, M., *Towards a Security Ontology*, IEEE Security and Privacy, 2003

[Dor97] Dorfman, M.S., *Introduction to Risk Management and Insurance*, Prentice Hall, 1997

[etr] http://www.etr.org/recapp/research/researchglossary.htm

[FISMA] http://csrc.nist.gov/policies/FISMA-final.pdf

[Fri01] Frisinger, A., *Improving the protection of assets in open distributed systems by use of X-ifying risk analysis*, Proceedings of the IFIP TC11 Sixteenth International Conference on Information Security, 2001

[Gal06] Galatonov, T., Ph.D. Thesis, to appear;

[Gal92] Galliers, R., *Information systems research; issues, methods and practical guidelines,* Henley on Thames Waller, 1994

[Gaz04] Gazendam, H.W.M., *Organizational Semiotics: a state of the art report*, Semiotics: A Global Information Bulletin, Volume 1, Issue 1, 2004

[Gaz05] Gazendam, H., Liu, K., *The evolution of Organizational Semiotics – A brief review of the contribution of Ronald Stampler,* To appear in: Joaquim Filipe & Kecheng Liu (Eds.). Studies in organisational semiotics. Dordrecht: Kluwer Academic Publishers, 2005

[Gua87] Guarro, S., *Principles and procedures of the LRAM approach to information systems risk analysis and management*, Computer and Security 6(6), 493–504, 1987

[Hal96] Halliday, S., Badenhorst, K., von Solms, R., *A business approach to effective information technology risk analysis and management*, Information Management and Computer Security, 1996

[Ham80] Hamilton, P., Kettell, A., *Business security – internal policing for management*, Associated Business Press, 1980

[Haw02] Hawker, A., *Security and Control in Information Systems*, RoutledGB, 2000

[Hol04] Holton, G.A., *Defining Risk*, Financial Analysts Journal, 2004

[Hoo04] Hooghiemstra, T.F.M., *Confidentiality chain*, National ICT Institute for Healthcare, the Netherlands, 2004

[Hoo05] Hoogervorst, J.A.P.*, IN4153 Lectures– Architectural Design of Business Systems*, 2005.

[iso] http://www.iso-standards-international.com/

[Jur05] Jurjens, J., *Secure Systems Development with UML*, Springer, 2005

[ker] http://web.mit.edu/kerberos/www/

[Kra72] Kraus, L., *SAFE: Security Audit and Field Evaluation for Computer Facilities and Information Systems*, AMACOM, USA, 1972

[Lie97] Liewo, J., Zheng, Y., *A formal model to aid documenting and harmonizing of information security requirements*, Information Security in Research and Business, Proceedings of the IFIP TC11 13th International Conference on Information Security, Copenhagen, Denmark, 1997

[Liu00] Liu, K., *Semiotics in information systems engineering,* Cambridge University Press, 2000

[Liu02] Liu, L., Yu, E., Mylopoulos, *Analyzing Security Requirements as Relationships Among Strategic Actors,* In: Proceedings of 2nd Symposium on Requirements Engineering for Information Security (SREIS'02). Raleigh, North Carolina, October 16, 2002.

[Liu02] Liu, K., Sun, L., Barijs, J., Dietz, J.L.G., *Modeling dynamic behavior of business organizations – extension of DEMO from a semiotic perspective*, Elsevier Science B.V., 2002

[Liu03a] Liu, L., Yu, E., Mylopoulos, J., *Security and Privacy Requirements Analysis within a Social Setting*, Proceedings of the International Conference on Requirements Engineering (RE'03). Pages 151-161.Monterey, California, September 2003.

[Lui03b] Liu, L., Yu, E., *Designing Information Systems in Social Context: A Goal and Scenario Modelling Approach*, Information Systems (journal), Vol.29, No.2. 2003. Elsevier Ltd. (revised and extended version of [CAiSE02])

[Mai02] Maij, E., Toussaint, P.J., Kalshoven, M., Poerschke, M., Zwetsloot-Schonk, J.H.M, *Use Cases and DEMO: Aligning Functional Features of ICT-infrastructure to Business Processes*, International Journal of Medical Informatics, 2002

[Mcd99] McDermott, J., Fox, C., *Using abuse case models for security requirements,* Proceedings of the 15th Annual Computer Security Applications Conference (ACSAC), 1999

[Mcg98] McGrow, G., *Testing for security during development – why we should scrap penetrate-and-patch*, IEEE Aerospace and Electronic Systems, 1998

[mic] www.microsoft.com

[Mos92] Moses, R., *Risk analysis and management*, Computer Security Reference Book, Oxford, Butterworth-Heinemann, 1992

[Mou96] Moulton, R., Moulton, M., *Electronic communication risk management: a checklist for business managers*, Computer and Security, 1996

[Mur84] Murine, G., Carpenter, C., *Measuring computer system security using software security metrics. In Computer Security: A global challenge (JH and Dougall EG, Eds)*, Finch Elsevier Science Publisher, Proceedings of the second IFIP International Conference on Computer Security (IFIP/Sec'84), 1984

[ORANGE] http://www.radium.ncsc.mil/tpep/library/rainbow/5200.28-STD.html

[org] http://www.orgsem.org/

[Ras01] Raskin, V., Hempelmann, C.F., Triezenberg, K.E., Nirenburg, S., O*ntology in Information Security: A Useful Theoretical Foundation and Methodological Tool*, Proceedings of the New Security Paradigms Workshop, New York. ACM, 2001.

[Ras02] Raskin, V., Niremburg, S., *Ontology in Information Security: A useful Theoretical Foundation and Methodological Tool* , New Security Paradigms Workshop (NSPW), 2002

[Sal83] Saltmarsh, T., Browne, P., *Data processing – risk assessment*, In: Advances in Computer Security Management (WOFSEY MM, Ed), Vol 2, pp 93–116, John Wiley and Sons Ltd: New York, 1983

[Sea69] Searle, J.R., *Speech Acts: an Essay in the Philosophy of Language*, Cambridge University Press, 1969

[sec] http://www.securityforum.org

[Sip05a] Siponen, M.T., *An analysis of the traditional IS security approaches: implications for research and practice*, European Journal of Information Systems, 2005

[Sip05b] Siponen, M.T., *Analysis of modern IS security development approaches: towards the next generation of social and adaptable ISS methods,* Information and Organization in press, 2005

[Soh05] Sohr, K., Drouineaud, M., Ahn, G., *Formal specifications of Role-based Security Policies for Clinical Information Systems,* ACM (Association for Computing Machinery) SAC (Annual ACM Symposium on Applied Computing), 2005

[Sta73] Stamper, R. K., *Information in Business and Administrative Systems*, Halsted Press, 1973

[Sta96] Stacey, T., *Information security program maturity grid*, Information Systems Security 5(2), 22–33, 1996

[Str93] Strens, R., Dobson, J., *How responsibility modeling leads to security requirements*, Proceedings of the 16th National Computer Security Conference, 1993

[Swa03] Swanson, M., Bartol, N., Sabato, J., Hash, J., Graffo, L., *Security Metrics Guide for Information Technology Systems*, NIST Special Publication, 2003

[Tan03] Tan, S., Liu, K., Xie, S., *A Semiotic Approach to Organisational Modelling using Norm Analysis*", UK Academy for Information Systems, 2003

[Tho94] Thomas, R.K., Sandhu, R.S., *Conceptual Foundations for a Model of Task-based Authorizations*, Proceedings of the 7th IEEE Computer Security Foundations Workshop, 1994

[Tru03] Trust, R., Kannan, K.P., *E-service: A New Paradigm for Business in the Electronic Environment*, 2003

[Wie02] Wieczorek, M, Naujoks, U., Bartlett, B., *Business continuity*, Springer, 2002

[wik] http://en.wikipedia.org/wiki/

[Woo87] Wood, C., Banks, W., Guarro, S., Garcia, A., Hampel, V., Sartorio, H., *Computer Security: A Comprehensive controls Checklist*, John Wiley and Sons, 1987

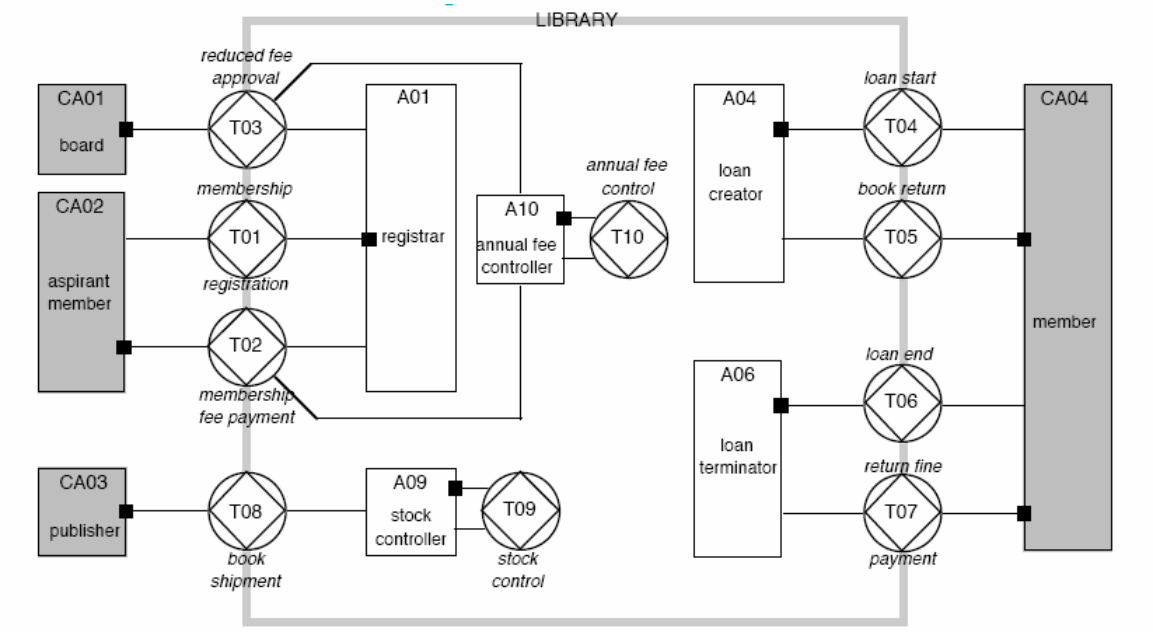
[Yin94] Yin, R.K., *Case Study Research: Design and Methods*, London : Sage, 2nd edition, 1994

Annex 1: DEMO diagrams for the Library case study

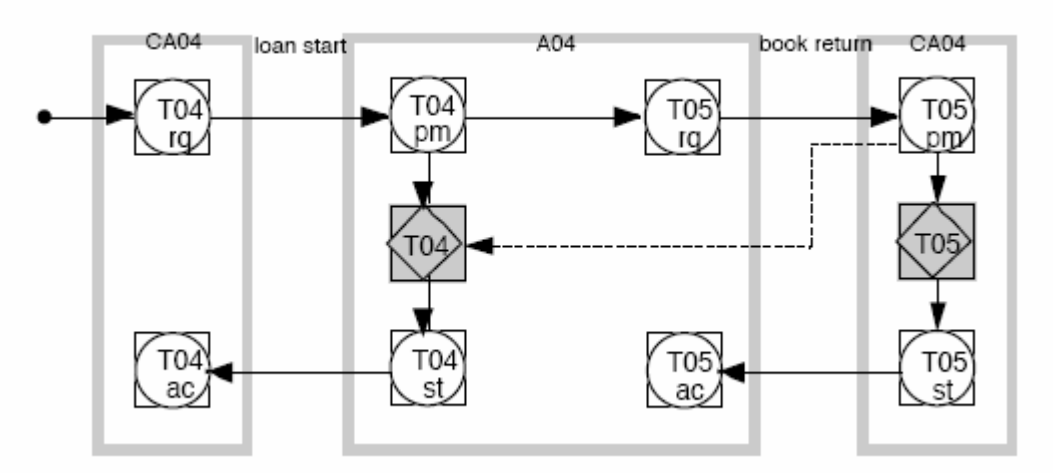
The library described hereafter is the small but autonomous public library of Delftown. In the building in which it is located, is a desk for lending books,called the out-desk, and a desk for returning books, which is called the indesk. The in-desk is occupied by Sanne and the out-desk by Tim and Kris on turn. There is a third desk, called the information desk, which is occupied by Lisa. At the information desk one can get information such as opening hours, loan rules, and membership fees. There is also a binder on Lisa's desk that contains the complete library catalog, sorted in several ways (on author, on category and on title). One can freely browse through the binder to find the book one is looking for. Next to that, one can ask Lisa any questions about the library, e.g. about the contents of the books in the catalog. The information desk also serves as the registration desk. Anyone who wants to be registered as member of the library has to apply with Lisa. She writes the data needed on a registration form. The requested data are: surname, first name, middle initials, city of residence, street name, house number, postal code, sex, date of birth, starting date of the membership, and annual fee. By default, the annual fee is the standard annual fee as determined by the library board. Exceptions may be made for people without means. In that case, Lisa applies in writing to the library board for the reduced fee, which is a symbolic 1€ per year. The applicant then has to fill out a form in which a specification of the income and expenses in the past calendar year are asked for. This form is attached to Lisa's letter. The registration forms regarding regular memberships are collected daily (after closing time) by Sanne who puts the data in the Library Information System (LIS) that runs on the only PC of the library. LIS automatically prints a membership card and an invoice for every new member. The invoice regards the remaining months of the current calendar year, including the current month. So, for example, if one registers in September, one has to pay 4/12 of the annual fee. Both the card and the invoice can be collected by the regular new members from the next day on at the information desk. One then also gets a letter of welcome, informing the new member about the library rules. Membership cards have a bar code on it representing the membership number. They are handed over to the new member after cash payment of the membership fee.

For members that apply for the reduced fee, the procedure is slightly different. They have to wait until they are informed in writing about the decision of the board. This is something Tim takes care of. As soon as he gets the decision of the board, he writes an according note and sends it by postal mail to the applicant. A copy of the note goes to Lisa. If the reduced fee is allowed by the board, Lisa takes the registration form out of her drawer and hands it over to Sanne, for processing at the end of the day. In case of a negative decision by the board, she inserts the form in the file of declined applications. People whose application has been declined, are supposed to have cancelled their original request for registration. They may of course register again, but then only as full paying members. The books that can be borrowed are put on shelves, and sorted on the category of the book title. There may be several copies of the same book title. Every such book copy is uniquely identified by a bar code. This code contains both the ISBN (International Standard Book Number) and the serial number of the book copy. If one wants to borrow a book, one has to take (a copy of) the book from the shelves and bring it to the out-desk. Tim or Kris will then scan the bar code on the membership card, as well as the bar code on the book. These data are automatically entered into LIS. The book is now considered to be lent to the member. No more than 5 books may be lent simultaneously to the same member. When one returns a book, one goes to the in-desk and hands the book to Sanne. She scans the book code, which is automatically entered into LIS. On the screen of her computer, she sees whether the loan period is exceeded or not. If it is, she also sees the fine that has to be paid. The person who returns the book has to pay the fine right away and in cash. After payment, Sanne marks the book in her computer as returned. If the loan period is not exceeded, she only enters that the book has been returned. Returned books are piled on a table next to Sanne. About every hour Lisa collects the pile and puts the books back on the shelves. While she is doing that, the information desk is temporarily unoccupied. Every month, the librarian decides which titles should be added and how many copies per title have to be ordered. She does so on the basis of the announcements of new books she knows of (by means of flyers of publishers but also by surfing on the web) and on the basis of analysis reports of the reading habits of the members that are provided by LIS. The librarian disposes of an annual budget for buying new books that is decided upon by the board of the library. An order is directed to one publisher, but it may regard a number of copies of a number of book titles. At the start of a new calendar year, Kris sends out invoices to all current members for the annual membership fee. Fees have to be paid in cash the next time one comes for borrowing a book. She also sends prolongation requests of reduced fee memberships to the board. Attached to them are statements of income and expenses over the past year that she has asked the applicants to produce. She deals with the decisions by the board in the same way as is done the first time of application.

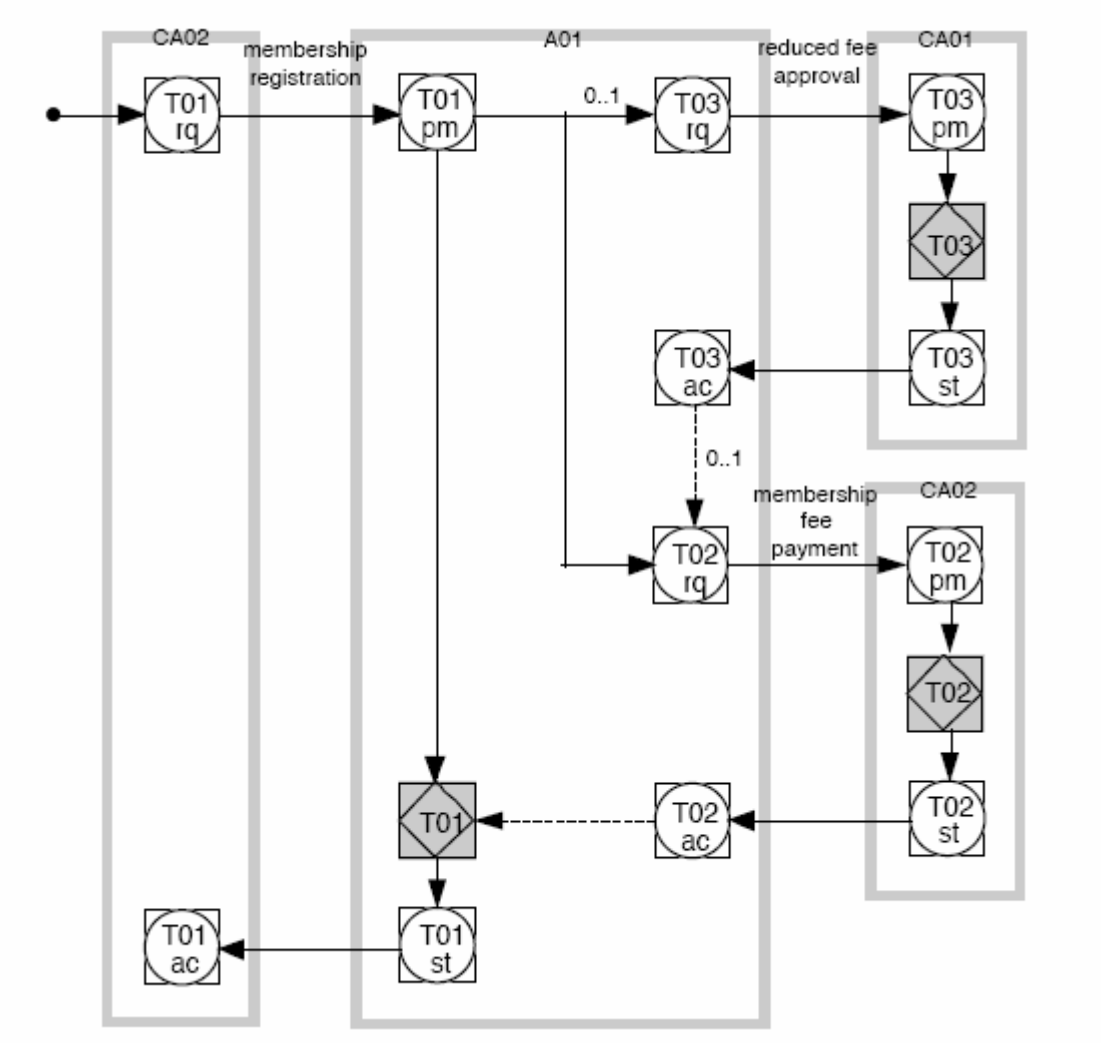
(excerpt from [Die05a])



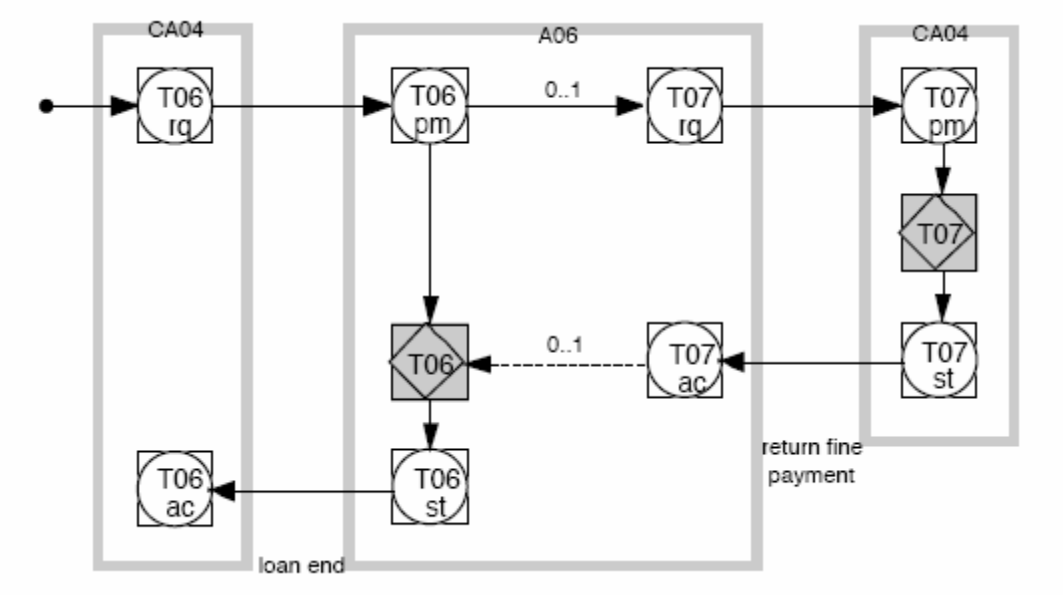
Annex 1 - 1: Complete Actor Transaction Diagram of a Library [Die05a]



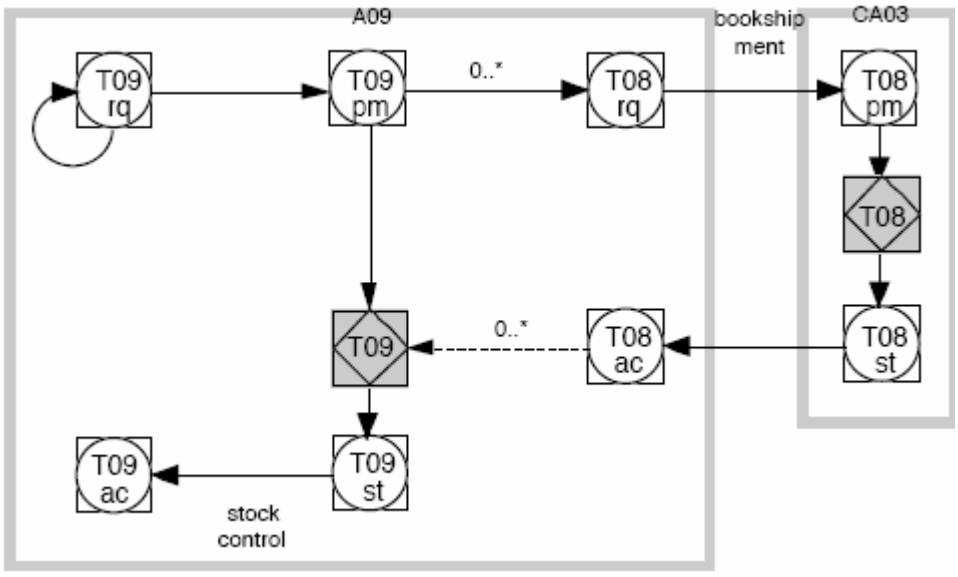
Annex 1 - 2: PSD for the process 2 of the library [Die05a]



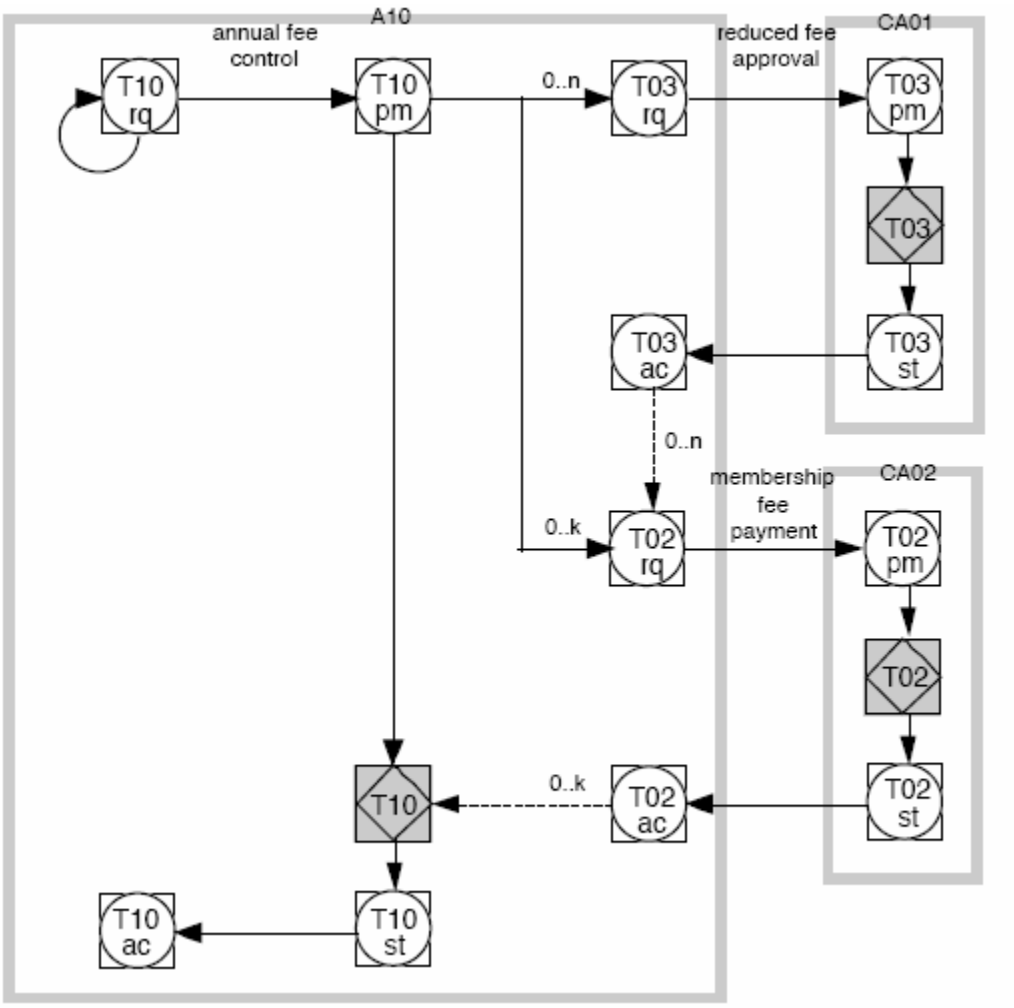
Annex 1 - 3: PSD of business process 1 [Die05a]



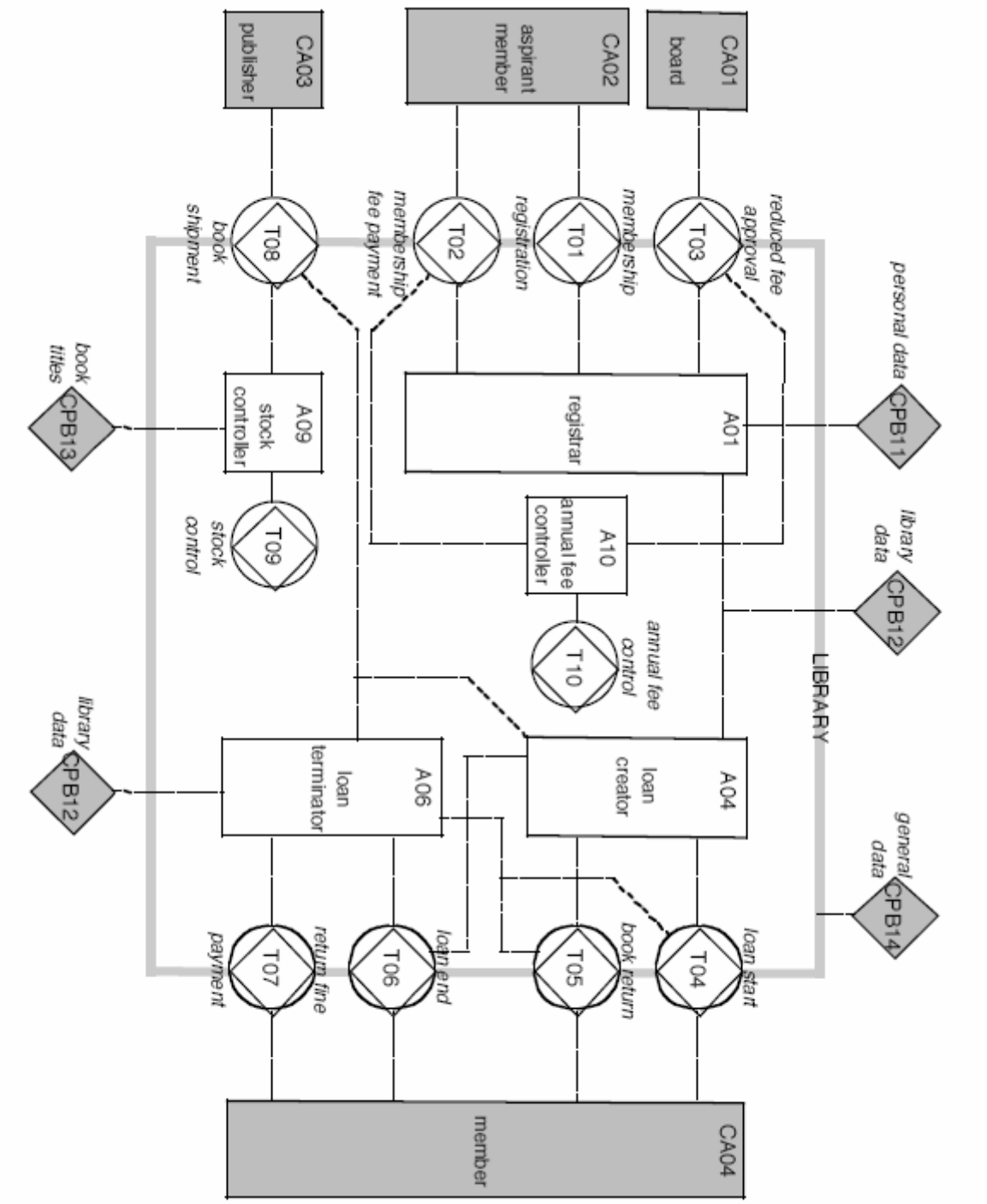
Annex 1 - 4: PSD of business process 3 of the Library [Die05a]



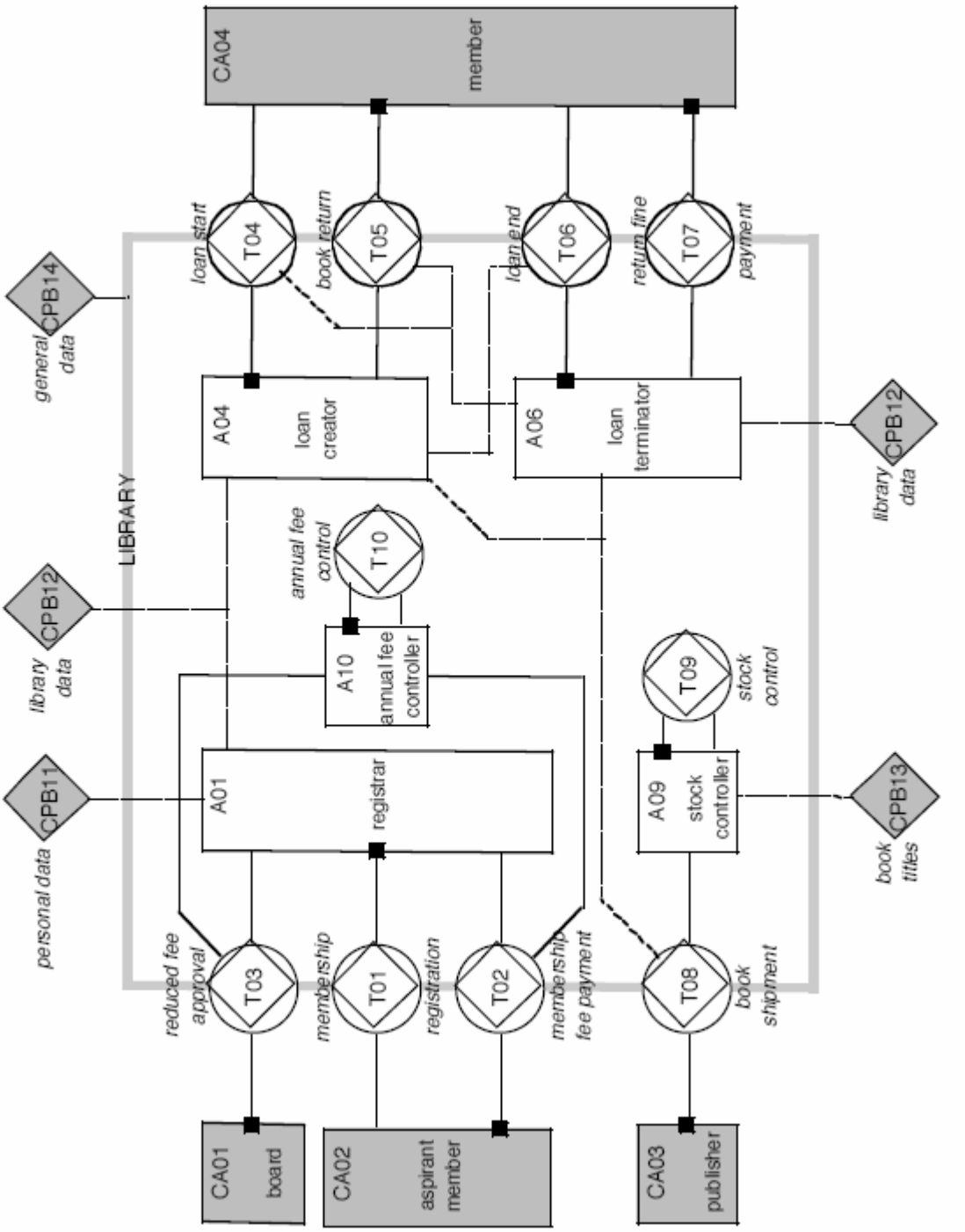
Annex 1 - 5: PSD of business process 4 of the Library [Die05a]



Annex 1 - 6: PSD of business process 5 in the Library [Die05a]



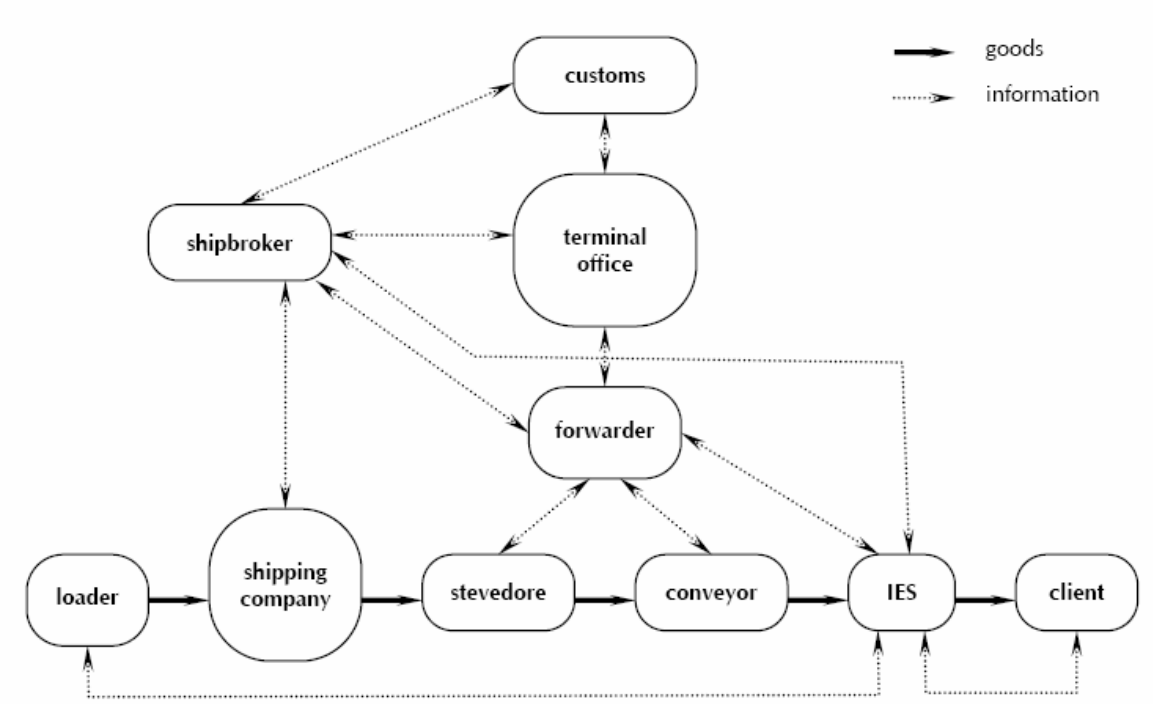
Annex 1 - 7: The Actor Bank Diagram for the Library case [Die05a]



Annex 1 - 8: The Organization Construction Diagram of the Library [Die05a]

Annex 2: DEMO diagrams for the IES case study

IES is a company situated in Baile Átha Luain with specialization in logistics. Its responsibility comprises the entire production delivery chain for its clients. For example, one of its tasks it to import customer electronics from the Far East countries on behalf of European Dealer companies, in containers, by ship. IES is responsible for the whole delivery process: from placing an order to delivering the product to the client. To do this, IES uses the services of different organizations; their cooperation scheme is as in Annex 2 - 1 [Gal06]:



Annex 2 - 1: Scheme of global organizational cooperation for the IES case [Gal06]

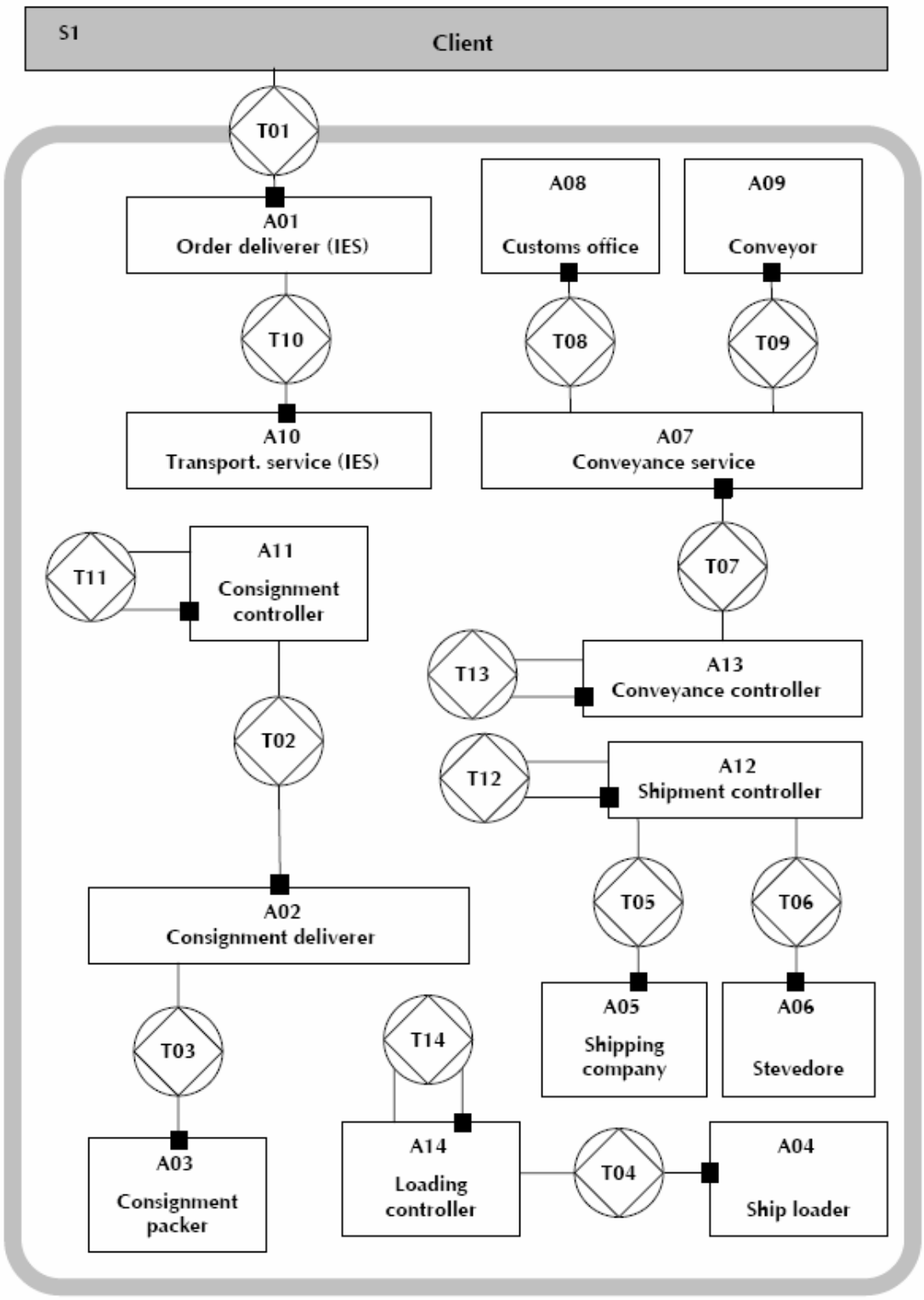
The schema goes as follows: the good goes from the 'loader', through the 'shipping company', through the 'stevedore' (container uploading organization), then through the '(overland) conveyor' which transports goods along the route Luimneach and Baile Átha Luain and than to the IES that has to do the temporary storage and the delivery to the client. The 'shipbroker' and the 'forwarder' have coordination roles whereas the 'customs' has the authorizing role. There is then the 'terminal office': this is about the coworkers in the territory of the port of the 'shipbroker', the 'forwarder' and the 'customs' [Gal06].

Further on, we will describe the way IES conducts its business as it is depicted in [Gal06]. IES takes the order of the client and than makes a demand to the supplier in the Far East (one example of client is a dealer organization). It is important to say that one order concerns just one item (i.e. one radio brand). Each day, IES takes all the demands from the clients and composes the orders for each supplier: one order contains one item. When a supplier receives an order, the 'loader' gets together all the corresponding merchandize and loads it onto containers in ships that belong to the 'shipping company'. The ship gets to Luimneach and 'stevedore' is the one that unloads it. Then the goods are loaded onto 'conveyor's' trucks and then transported to the IES where they are again unloaded. The delivery service of IES sees that the goods are delivered to the 'clients'.

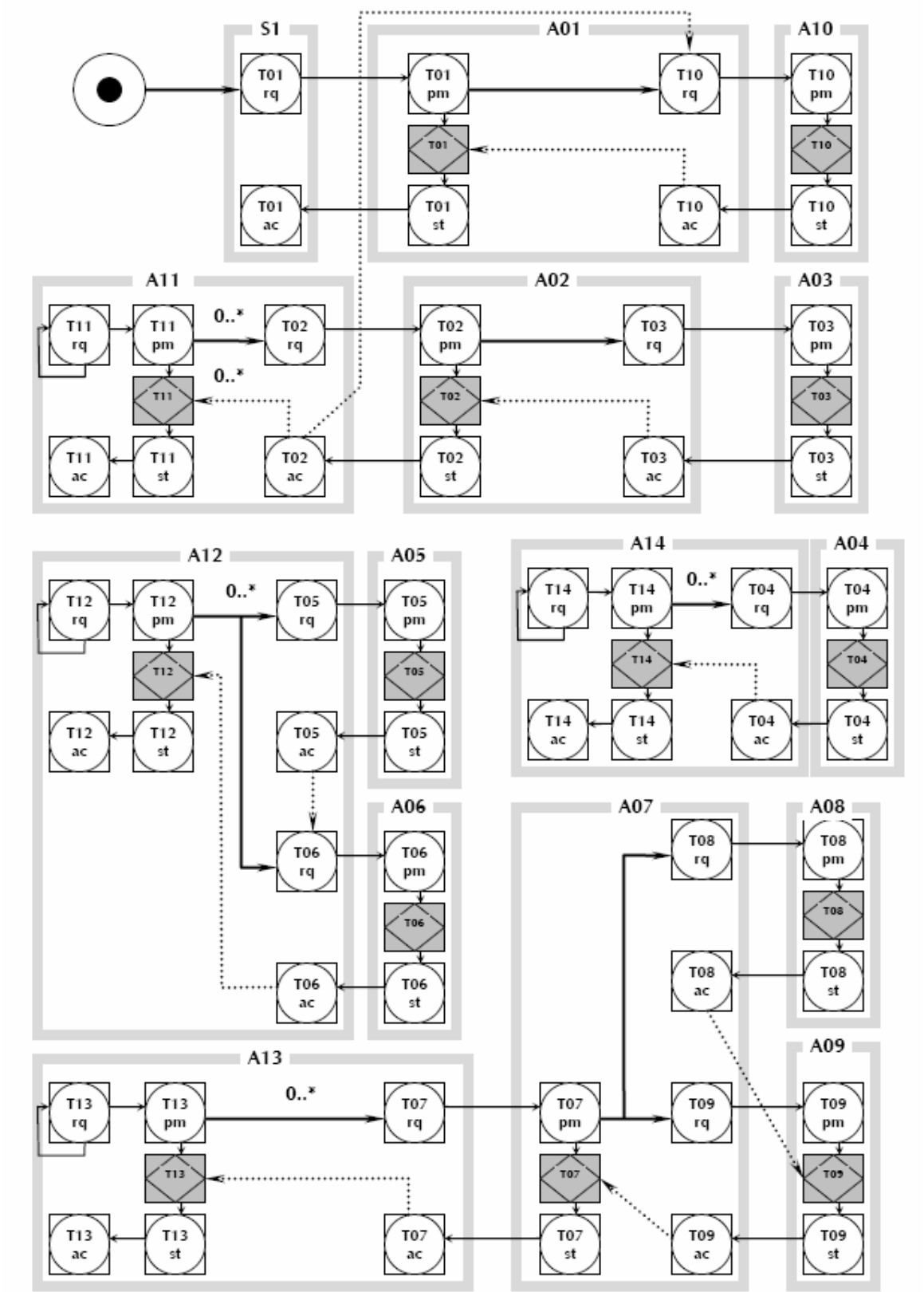
The following basic activities can be identified for the parties involved:

- **IES**: is the initiator of the shipping operations – it verifies and, if necessary, corrects the cargo documents that were sent by the 'loader', and then passes the documents to the 'forwarder', receives the goods and delivers them to the client; its responsibilities also include planning the operations for the container transport, and providing the 'forwarder' with the time, the place and specific data for the containers to be transported;
- **Shipbroker**: is the shipping company responsible for the ship and cargo of the goods – it informs the 'customs', IES and the 'forwarder' about the cargo' specific data and the expected arrival time of the ship; when a container is empty (unloaded) and ready for further use, it is the one that needs to report this thing;
- **Stevedore**: is the one responsible for the unloading of the containers – it takes care of all the time and location arrangements with the shipping company for the unloading; also takes care of the storage of the containers (on its own basis) until they undertaken by the 'conveyor'
- **Forwarder**: is the one responsible for the transportation of the containers from 'stevedore' to IES – together with the 'conveyor' it arranges when the containers should be collected and delivered; to do this, it informs the 'conveyor' about all the necessary data concerning the collection of the containers from 'stevedore';
- **Terminal office**: it is the office that lodges all the offices on the territory of the port corresponding to the customs, shipbrokers and the forwarders.

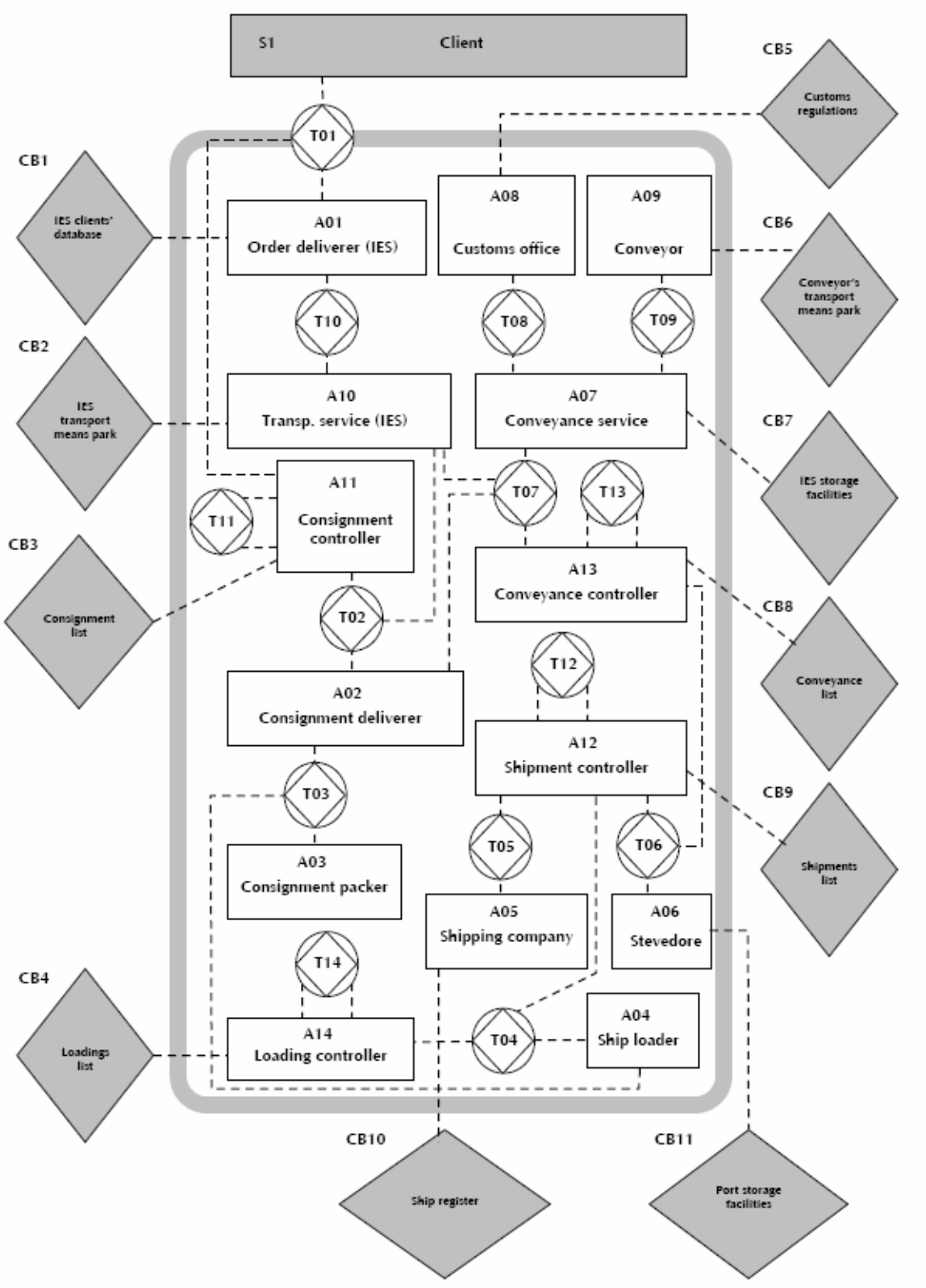
For further more detailed information, we refer the reader to the DEMO diagrams of the case which are presented next in this annex.



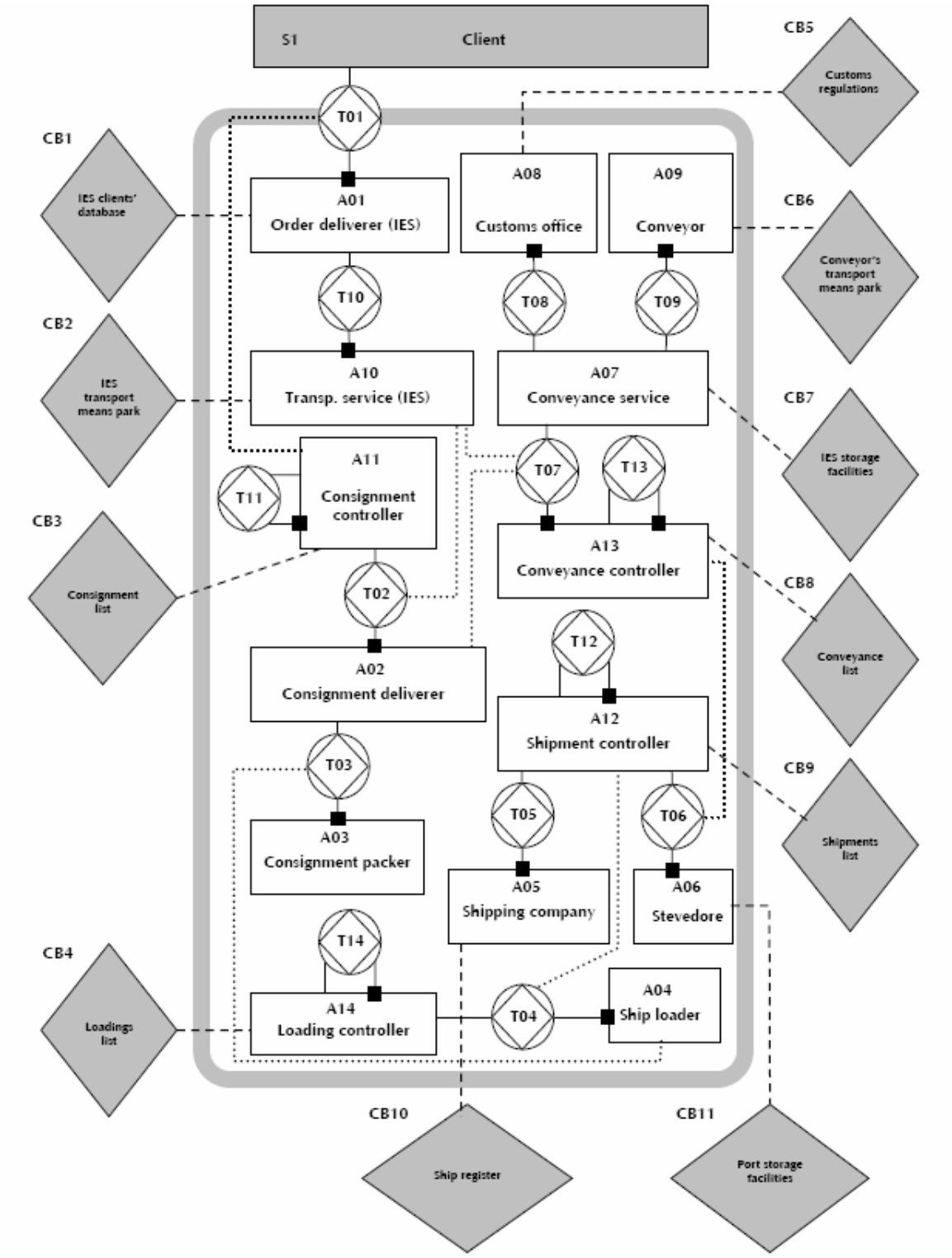
Annex 2 - 2: The ATD for the IES case [Gal06]



Annex 2 - 3: The Actor Structure Diagram for the IES case [Gal06]



Annex 2 - 4: The Actor Bank Diagram for the IES case [Gal06]



Annex 2 - 5:The Organization Construction Diagram (OCD) for the IES case [Gal06]