DESIGNING AND MAINTAINING TRUSTWORTHY ONLINE SERVICES

Christer Rindebäck

Blekinge Institute of Technology
Licentiate Dissertation Series No. 2007:08
School of Engineering

Designing and maintaining trustworthy online services

Christer Rindebäck

Blekinge Institute of Technology Licentiate Dissertation Series
No 2007:08
ISSN 1650-2140
ISBN 978-91-7295-120-4

Designing and maintaining trustworthy online services

Christer Rindebäck

Department of Interaction and System Design
School of Engineering
Blekinge Institute of Technology
SWEDEN

Department of Interaction and System Design
School of Engineering
Publisher: Blekinge Institute of Technology
Printed by Printfabriken, Karlskrona, Sweden 2007
ISBN 978-91-7295-120-4

Abstract

Trust and trustworthiness are two notions that have been discussed extensively in the computer science community, e.g. trust in online banking services. We argue for a broad view on trust, namely trustworthy behavior of online services. We propose solutions enabling online service developers to reason about, and deal with issues of trustworthy online services, from concerns to actual implementations, and assessments. The view on trust in this thesis involves viewpoints on what stakeholders can have trust in, and the need to exhibit and suggest trustworthiness in online services. Trustworthiness and other relevant theories are also discussed. Three main results supporting design and maintenance of trustworthy online services will be introduced.

First, a trust framework in the context of online services is introduced, specifying a number of concepts that enhances and clarifies how trust can be addressed. The framework enables an informed analysis, implementation, and assessment of solutions to trust issues based on identified trust concerns.

Secondly we present how the concepts of the framework can be interconnected. The concepts enables us to reason about stakeholders trust concerns in relation to deployable solutions called trust mechanisms that are implemented in order to exhibit proper signs suggesting trustworthiness. These signs, we argue, serve as input for stakeholders trust assessment. The interconnected framework opens up for a discussion on how deployed solutions in an online service correspond to certain stakeholders trust concerns.

Finally a tool for online service designers is presented, the trust management life cycle. This is an approach enabling an informed design practice that emphasizes on a trustworthy design of online services. The use of the cycle is illustrated by the use of a deployed online service.

Acknowledgements

Many persons have contributed to the work with this thesis and it would have been impossible for me to present the work here without their assistance and appreciation. First of all I would like to thank my main supervisor, professor Rune Gustavsson for his constant support. Rune also gave me the opportunity to enroll my path as a research student and encouraged me to look into the domain of trust and trustworthiness. Without his insights, ideas and supervision this work would not have been possible. I would also like to thank my assistant supervisor, professor Paul Davidsson for his comments on my work.

The members, past and present of the SoCLab group which currently consist of professor Rune Gustavsson, Jenny Lundberg, Kerstin Ådahl, Björn Ståhl, Dr. Per Mellstrand and Dr. Martin Fredriksson. I'd also like to thank Lousie Östlund and Patrik Brandt from the University of Kalmar.

Furthermore I would like to thank the members in the work practice and technology research group. Also thank you, Dr. Berthel Sutter for the discussions and comments on activity theory. During the preparation of this thesis I received valuable assistance from Anna Harding and Stefan Johansson.

Other staff members at the School of Engineering who make my work inspiring and interesting also deserve to be mentioned here, thank you all.

I'd also like to thank my friends 'at home' for their support and encouragement throughout this work.

Finally I would like to thank my Mother, Father, and my Sisters, Pernilla and Sophia, for their support during my work with this thesis.

Christer Rindebäck

Table of Contents

Part I

Motivation

Introduction

1 Background

Building and maintaining trustworthy online services is a challenging task. This is illustrated in many ways, among others in the European Union's 7:th framework research program ICT - Information and Communication Technologies where trust is one area that is mentioned frequently when it comes to the design of future ICT technologies [1]. Throughout the history of online service creation and use, different problems have evolved over time that has its roots in concerns related to a person's perception of aspects such as security, responsibilities, correctness, and identity of online service providers. The value of online sales are estimated at approximately $31.5 billion in the United States alone during the first quarter of 2007 [2]. There are numerous vendors offering online services today. In some cases established companies initializes online presence by means of online services. In other cases start-ups provide services where the service itself is the motivation for company formation.

On online auction sites such as e-Bay or Swedish Tradera[1] goods are exchanged between persons based on reputation and security features implemented by service providers. These services affect habits and manners of doing business in the comfort of users' own homes. Today the abovementioned actors are well established and well known among Swedish internet users. Needless to say, the perception of the services offered differs among end users of services. In some cases users' consciously reflects about whether a particular service is to be used or not, for instance we may or may not decide to use a service due to concerns about the quality of the promoted services. There may also be issues related to the price of the service, or the involved technology. In some cases the service is simply not perceived as needed at present, but the user may reconsider it later.

Not only may issues be connected to the functionality and quality of the service as such, but questions frequently asked by customers include: Is it safe to initiate a credit card transaction with the online vendor?, and is the online vendor able to retain a good service level? Furthermore questions such as: what happens if I don't receive the goods from y?, or who is responsible if there is a legal dispute? are raised. The first set of questions can be described as preferential, that is there may not be a need for the service or product at the time, or the potential buyer may not be able to pay for a service at the time. The reasons why a service is used or not is thus multi-dimensional.

This thesis isn't targeted towards marketing issues or purchasing patterns, it is dealing with issues which we denote as trust concerns. The theme for the thesis is how we can deal with trust issues during the life span of online services. This includes how concerns can be identified and addressed. For instance, if potential online service users have concerns about online payment security, how can their concerns be reflected in an online service design? There are approaches to deal with trust in online services already

[1] Tradera is owned by e-Bay and targeted toward the Swedish market

in place such as [3]. Also numerous studies has been made to identify important factors in e-commerce applications [4, 5]. The approach in this thesis emphasizes on factors and concerns expressed by (or related to) stakeholders and how to deal with these when engineering sustainable trustworthy online services. The software intensive systems put in place to provide online services should be based on relevant factors, typically specified by a set of requirements. We claim that the implementation of concerns identified into deployable solutions must be done with uttermost care in order to avoid trust issues throughout the life cycle of an online service.

1.1 Internet Banking Example

In order to further elaborate on trust and concerns in the context of online services we turn to the area of Internet banking. In studies it has been claimed that users may not trust Internet banks. In Sweden it is common to carry out banking tasks by using online based services, often referred to as Internet banking. Most major banks as well as new so called niche-banks encourage customers to use the provided online banking services to carry out tasks related to banking. This includes payment handling, transfers, stock-trading, and loan applications. Given the sensitive nature of most banking activities, that is money is involved, there are associated risks. Money can get lost if the systems aren't developed in the right way and handle e.g. account transfers correctly.

Internet banking enable users to affect transactions in a bank's the backend system by giving commands from a computer connected to the Internet banking service. In order to avoid unauthorized transfers Internet banks are typically equipped with security measures to restrict access to certain information. Encryption and authentication are two techniques uses to protect the access to sensitive information and unauthorized access to the online banking services. Most Swedish Internet banks have access-based control solutions where one-time access codes are used in combination with a login id and password. The one-time access code is deemed as used after one login or authentication. One time access codes are also often used to sign payments and transfers whilst already logged in. These actions try to deal with a number of concerns that end-users, that is banking customers, using the Internet banking service may have. Examples of end-users opinions are: "I'm concerned that my personal financial information can be read by unauthorized parties" or "My money can be transferred to the wrong account".

Internet banks can and are taking precautions to anticipate certain trust concerns. What also should be considered is the fact that banks *themselves* have trust concerns e.g. "People may try to access confidential information" or, "Criminals may attempt to transfer money by using Internet banks". Thus different actors may have different concerns that may be or not be dealt with. Some concerns Internet banking service providers may be aware of, and some not. Some are relevant justified concerns, and some are not. Some are concerns today but not tomorrow. Some concerns may be more important for a certain actor than another and the disposition of importance may vary greatly. However, we suggest that trust concerns should be the starting point to find out what qualities that should be deployed at a particular point in time in order to address issues related to trust and trustworthiness.

It is important to point out that concerns may or may not be based on rational premises. This resembles ideas from risk assessment research where studies have shown

that the perception of risk often is based on non-rational ground [6]. Rational or not, trust concerns can be used by online service designers in order to identify appropriate mechanisms that deals with the identified concerns. In the case of Internet banking for instance a number of concerns related to aspects of privacy and security was highlighted in the example above. In order to resolve issues a set of mechanisms (encryption and authentication) has been described, and implemented with the intention to address the concerns expressed by the involved stakeholders. Ideally the presence of mechanisms is visible and perceived by the stakeholders as viable solutions as a whole, or as part of their trust concerns. The same reasoning can be applied in other domains such as e-health which will be illustrated later on in the thesis.

To summarize, different actors (or stakeholders) can have different concerns regarding an online service. In the case above concerns regarding privacy needs to be balanced against the need of the bank to provide a service based on financial integrity. These concerns can to some extent be solved by using the same class of solutions, e.g. encryption which tackles both financial security and privacy concerns. However, some mechanisms may be more targeted towards certain concerns e.g. privacy seals.

1.2 A Short Introduction to Trust and Trustworthiness

The view on trust that will be used during this thesis is that it is subjective and depending on assessments and expectations of humans which is used in situations of risk where there is a lack of complete information. An expectation is according to Barber [7]:

> "the meanings actors attribute to themselves and others as they make choices about which actions and reactions are rationally effective and emotionally and morally appropriate."

Whatever the lack of trust is dependent on, we may need to address these issues in order to enable a more positive trusting experience. Deeper perspectives on trust are highlighted in the chapter *Theoretical Background* starting on page 13. In situations of risk we may choose to act or not to based [8] on a trust assessment. This assessment in turn may be based on rational or non-rational(emotional) grounds based on correct and non-correct information and perceptions [6, 9]. Furthermore self confidence is a factor affecting trust [10, 11]. Trust has a practical function in daily life since it is used to reduce complexity in decisions in situations involving risk [8].

The complexity of factors and subjective perceptions makes trust a challenging area to target. Although problems have been identified and attempts are made to tackle them there is no principled way to address issues such as trust in the context of software intensive systems. In requirements engineering non-functional requirements are discussed but are often considered to be problematic to handle. The main purpose with this thesis is to present an approach that enables an informed discussion on trust properties in the context of online service development, deployment and assessment. An approach which also emphasizes on how a quality such as trust can be addressed thoroughly throughout the life cycle of online services. Today online service providers are not fully aware of the complexity of trust issues and hence are unable to fully cope with the issues raised by end users and online-service providers. Studies dealing with issues related to trust,

confidence and web site credibility indicate that much work has and can be done in order to target issues identified by stakeholders such as end users and online service designers.

Giving attention to trust in online service development doesn't necessarily lead to increased costs; on the contrary it may turn out to be the opposite. If no concerns are present there may not be a need to implement certain solutions to address non-existent trust concerns (i.e. implement unnecessary features for the sake of trust)[2]. Since trust is a human property we want to distinguish the work presented from what is known as computational trust. The intention of the field of computational trust is to create a trusting behavior between computational entities, e.g. software agents [12]. The viewpoint presented throughout this thesis is that human trust can't be built into systems; however, actors can suggest that they are trustworthy [13].

To summarize the view on trust proposed here is as follows: Trust is based on human subjective assessments influenced by factors such as experiences, self confidence and perceptions of risk. In order to facilitate trust assessment online services should be designed in a way that fosters proper implementations with respect to trust. This initial discussion will be further elaborated on in the next chapter, *Theoretical Background* and throughout the thesis.

2 Challenges and Motivation

Trustworthy online services is only one area that is interesting from the perspective of the development of software intensive systems. Qualities of systems in use has lately been driving the research within human computer interaction, and computer supported cooperative work. In work by Mellstrand [14] the focus is on the execution of software in run-time trying to intercept and address security issues when the system is actually being executed. The reasoning behind this is that it is simply not possible to design and implement "perfect" software. This is interesting since many software developers try to accomplish something perfect, however, it isn't possible to cover all potential uses and loopholes [14]. This resembles probably the trusting experience. It is not possible to design a perfect trusting experience but rather an online service where trust can be addressed properly throughout the life-span of the online service. Stakeholders' perceptions of what is trustworthy will with very little doubt change over time due to circumstance both within and without control of online service designers. Trust in-use is what ideally would be the case not only upon delivery but throughout the use of the online service. The following can be stated as the main problem domains upon which this thesis is grounded:

2.1 Problem/Challenge area 1: the Problem of trust

Lack of trust has been highlighted by some analysts as one of the key issues to online success. However, no turn-key solution is present in order to deal with trust in the context of online services. Not only is the notion as such complicated to understand and

[2] However, it is important to point out that trust is just one quality of a system. Other issues such as stability, laws and regulations may require a certain solution to be deployed even if it isn't perceived as a concern.

grasp, it is also a non-functional quality that is hard to assess and to describe. Furthermore since trust is about risks it is typically connected to privacy and security issues. E.g. by enforcing privacy policies and encryption it is believed that sensitive information with larger certainty can be kept away from wrong hands.

In the global economy where users are online and present new, and not yet known concerns may occur. Cultural differences and concerns about identities of online service providers are examples of issues that may need attention in online settings. Seals of approval, privacy policies, certification programs etc. has been proposed and implemented but often there is no explicit study on how and in what way trust is address and in what manner the solutions are implemented and deployed in the online-service context. We claim that there is a need to better understand trust in the context of online services in order to enable an informed trust design in order to better discuss the success and failure of these efforts. To summarize, we denote this challenge as *the problem of trust*.

2.2 Problem/Challenge Area 2: Designing appropriate non-functional qualities into software-intensive systems

Trust is not a function that can be built into a system [15] just like a button or spell checker. It is a non functional quality that will change throughout the life-cycle of an online service. Today's development methodologies and approaches are in general based on the assumption that we can describe and specify all components of a system formally before it is constructed. E.g. we need a user interface and a database connection pool that can handle 250 requests per second. Typically most systems involving software today are software intensive systems (SIS). A SIS is any system where software contributes essentially to the design, construction, deployment or evolution of a system [16]. A car is a SIS just like the desktop computer or online services running on any computational device. A SIS should ideally be trustworthy for all users when it is to be used, just like a safe car should be safe not only when picked up from the dealer but throughout the life cycle of the car. When designing as SIS we typically anticipate the following four steps:

1. *What is needed?* – A project that decides that something needs to be addressed can be initiated based on insights gained from economical calculations, ethnographical studies of a workplace, investigation of workflows, and project development, or any other activity intended to understand or affect issues in an existing or potential real life setting. The decision to continue to develop some kind of system e.g. a SIS is based on the fact that there is a willingness to devote resources to carry out a project.
2. *What should be built?* – Here the result based on the domains investigated in the previous step is investigated. Decisions needs to be made on exactly what should be built and developed and a precise investigation on what should be build is made and formulated into an idea about what should be built, ordered and developed.
3. *How should we build it?* – Based on what we need to build we must now formulate how we should actually construct and implement the intended system. This step also include best practice and other process related issues. Methodologies and approaches on how to effectively managing the process of software development

including management, programming paradigms, patterns etc. has to do with this question.

4. *How do we cope with change?* – An often underestimated aspect of system development is how we should cope with changes. The technical term *maintenance* is often used. In mechanical systems maintenance is needed due to physical wear and tear. Software is a binary representation that is exactly identical for each copy of a piece of software [14] that won't suffer from wear and tear. Instead software gets old due to outdated hardware, new requirements on interoperability, changes in laws and regulations, new business needs etc.

2.3 Research questions

The research questions below are based on the problem areas discussed above and are as follows:
Research Question 1 (RQ1): What concepts should be part of a trust framework for online services?
Research Question 2 (RQ2): How are the concepts of a trust framework interconnected and used in a methodology?
Research Question 3 (RQ3): What tools can be designed to support an informed trust design and maintenance of online services?

We will return to these questions in Part 4 of this thesis where the results will be discussed.

3 Research Approach

The work in this thesis is partly based on experiences gathered through the Alfebiite Project[3] where we participated with the implementation of a Prolog-based application supporting trusted delegation of work tasks in health care settings. This work is presented in Appendix A (p.115) and contributed to a broader view on both trust and complex services. As a fact we have also been working with other health care scenarios. Interviews and discussions has taken place with municipality workers giving home health care in West Sweden during the implementation of a new mobile health care system, however, the company providing the system was put into bankruptcy during the early stages of the implementation. In general we have a qualitative view on research where unique settings can provide insights that can be further investigated and serve as a ground for more general ideas. The results in this thesis is thus based on such material but the intention is to further investigate the general applicability of the results presented.

4 Contributions of Thesis

The main contributions in this thesis is concerned with two result areas. This includes both a theoretical part mainly concerned with a deeper understanding for how trust is

[3] A Logical Framework for Ethical Behaviour Between Infohabitants in the Information Trading Economy of the Universal Information Ecosystem IST-1999-10298.

related to online services. To illustrate how and in what way the thesis contributes to the understanding of trust and online services is hard at this point. However, the results and contributions enables an informed discussion on trust in online services, not only from the perspective of problem formulation and problem statements; the contribution also enables a discussion regarding issues of implementation, e.g. how should we deploy a set of mechanisms that encompasses the identified problems?

4.1 Trust and online-services

The framework presented in paper 2 contains two contributions that are of major interest throughout the thesis. First of all we introduce a condensed view on trust categories and objects of trust. Based on this view the trust framework is introduced giving a foundation for how to address the problem of trust in structured manner. The trust framework outlines a number of notions e.g. the context, trust concerns, trust aspects, trust mechanisms and trust signs. These are all important concepts in order to enable an informed trust design of online services. The framework also introduces human actors into the context which is important since the intention with an informed trust design is to address human trust concerns.

4.2 Methods and guidelines

The methods and guidelines we have explained and outlined in this thesis can be used to further elaborate on and understand the problem of trust in online services. Not only will the framework introduced in paper 2 give an overview of the theoretical foundations, it also give online-service designers tools to analyze and reason about trust in conjunction with online service development in general. This work is further explained and discussed in the preceding paper 3 where the trust management cycle is introduced. The trust management cycle illustrates the process to make practical use of the more theoretical description presented when discussing the framework. Paper 4 further elaborates on methodological issues for online service development.

5 Structure of the thesis

This thesis is divided into parts. Below we outline the four different parts.

5.1 Part 1: Motivation

This section contains introduction to the thesis, and background information including a discussion on trust, trustworthiness and requirements engineering and related topics. The next chapter in this thesis, *Theoretical Background* is ending this part.

5.2 Part 2: Included Papers

Four papers are included in this thesis and are in part two *Included Papers*. Table 1 illustrate which of the following categories the papers are concerned with:

- *Trust/Trustworthiness* is where discussions on trust and trustworthiness is more deeply investigated and introduced in the context of online services.
- *Framework* denotes contributions to the trust framework which has been developed. Practice illustrates how the Framework can be applied in the context of online services.
- *Management* denominates that emphasis has been put on the trust management model developed.
- *Validation* presents indications that the introduced framework and practice can be used in order to discuss online services from the perspective of trust and trustworthiness in a beneficial way.

Topic / Paper	Trust	Framework	Practice	Management	Validation
Paper 1: Computational ecosystems in home healthcare	X				
Paper 2: Why trust is hard: Challenges in e-mediated services	X	X			
Paper 3: Design and Maintenance of Trustworthy e-Services: Introducing a trust management cycle			X	X	X
Paper 4: Functional versus Non-Functional Requirements Considered Harmful			X		

Table 1. A matrix illustrating where a number of concepts are discussed throughout the thesis

The papers included in this thesis are in chronological order as follows:

1. Gustavsson, R., M. Fredriksson and C Rindebäck (2001). Computational ecosystems in home healthcare. Social Order in Multiagent Systems. C. Dellarocas and R. Conte. Boston, Kluwer Academic Publishers. 2: 201–220. - This paper investigates information technology challenges in two health care contexts. Two scenarios are used to describe and reason about the material. The notion of institutions is discussed as well as computational ecosystems. In this context operationalizations of trust are also discussed.

2. Rindebäck, C. and R. Gustavsson (2005). Why trust is hard - Challenges in e-mediated services. In: Trusting Agents for Trusting Electronic Societies: Theory and Applications in Hci and E-Commerce: 180-199. – A framework for trust and e-Services is outlined and introduced. We present a view on trust based on who/what to trust and a set of trust dimensions. Home Health care is used as an applicatoin area to illustrate and discuss trust issues.
3. Rindebäck, C. and R. Gustavsson (2006). Design and Maintenance of Trustworthy e-Services: Introducing a trust management cycle. Second International Conference on Web Information Systems and Technologies, Setubal, Portugal, Insticc Press. – In this paper a trust management life-cycle is introduced to support a informed continuous trust design throughout the life-span of an online service. A scenario is used to illustrate how the life-cycle can be applied.
4. Gustavsson, R., J. Lundberg, C Rindebäck, and K Ådahl. (2007). Functional versus Non-Functional Requirements Considered Harmful. The proceedings or the 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing, Las Vegas, NV. p.95–101. – In this paper we investigate the relationship between non-functional and functional requirements and discuss why this division may be harmful. A number of tools that enable online service designers to reason about important qualities and challenges in relation to the services are introduced.

5.3 Part 3: Conclusions and Further Work

Section three, *Conclusions and Further Work* discusses the results and addresses future research directions based on the work presented in this thesis.

5.4 Part 4: Appendix

Finally, part four, *Appendix* presents work done to illustrate the implementation of trustworthy tools for doctors, nurses, and nursing assistants. The tool described is developed in Prolog and addresses issues regarding delegations of tasks based on formal warrants for specific task groups e.g. medication. The application discussed in the appendix has served as a basis for inspiration for the work described in this thesis.

References

1. European Commision: Ict - information and communication technologies: A theme for research and development under the specific programme cooperation implementing the seventh framework programme (2007-2013) of the european community for research, technological development and demonstration activities. Technical report, European Commision (2007)
2. U.S. Department of Commerce: Quarterly retail e-commerce sales: 1st quarter 2007. Published on the US Census Berau Webb (2007)
3. Egger, F.N.: From Interactions to Transactions: Designing the Trust Experience for Business-to-Consumer Electronic Commerce. PhD thesis, Technische Universiteit Eindhoven (2003)
4. Cheskin Research and Studio Archetype: Ecommerce trust study. Technical report, Cheskin Research Studio Archetype (1999)

5. Jones, S., Wilikens, M., Morris, P., Masera, M.: Trust requirements in e-business. Communications of the ACM **43** (2000) 80–87
6. Slovic, P.: The perception of risk. Risk, society, and policy series. Earthscan Publications, London ; Sterling, VA (2000)
7. Barber, B.: The logic and limits of trust. Rutgers University Press, New Brunswick, N.J. (1983)
8. Luhmann, N.: Trust and Power: two works. John Wiley, Chichester (1979)
9. Bacharach, M., Gambetta, D.: Trust as type detection. In Castelfranchi, C., Tan, Y.H., eds.: Trust and deception in virtual societies. Kluwer Academic Publishers, North Holland (2001)
10. Lee, J., Moray, N.: Trust, control strategies and allocation of function in human machine systems. Ergonomics **35** (1992) 1243–1270
11. Lee, J.D., Moray, N.: Trust, self-confidence, and operators adaptation to automation. International Journal of Human-Computer Studies **40** (1994) 153–184
12. Marsh, S.: Putting trust into e-commerce - one page at a time. In: Proceedings of the Fourth International Conference on Autonomous Agents (Agents' 2000). (2000) 73–80
13. Sisson, D.: e-commerce: Trust & trustworthiness (2000)
14. Mellstrand, P.: Informed System Protection. PhD thesis, Blekinge Institute of Technology (2007)
15. Friedman, B., Kahn, P.H., Howe, D.C.: Trust online. Communications of the ACM **43** (2000) 34–40
16. IEEE-SA Standards Board: IEEE recommended practice for architectural description of software-intensive systems. Standard Description 1471-2000, IEEE (2000)

Theoretical Background

The theoretical background is divided into sections that in turn relates to one or more of the posed research questions. Table 1 describes how the questions and content are connected.

Research Question/Area	Trust	Online Service Design	Activity Theory
RQ1	X	X	
RQ2		X	
RQ3		X	X

Table 1. The theoretical concepts of this chapter and their relation to the posed research questions

Trust and related concepts are discussed in section: 1 - *Trust and Trustworthiness* where the current state of trust research, theoretical issues and a number of definitions are presented. In section 2 - *Designing Online Services* the online service concept and approaches to design and maintenance of online services will be discussed. Finally in section 3 - *Activity Theory* the activity theory framework will be introduced.

1 Trust and Trustworthiness

The word trust has been used in everyday contexts where statements such as: “I don’t trust my car” or “my neighbors aren’t trustworthy” are used to describe some kind of personal attitude towards an object, organization or human actor. There is a multitude of trust definitions where the meaning are defined differently [1]. Furthermore what trust can be attributed towards and when it is appropriate to use terms such as trusted and trustworthy have different meanings depending on the definer’s context. In order to create an online service many organizations, computers, software, policies, humans and companies may contribute to the creation and delivery of the service. From the perspective of human actors e.g. end-users the question if the service is to be trusted or not may be posed and considered as an important factor in order to decide whether the service will be used or not. Trust has been highlighted as a major challenge in online contexts cf. [2] and has been suggested to be a way to reduce complexity when making decisions [3]. Many studies has been done indicating that a high level of trust increases the acceptance of online services. This approach can be seen in work on e-commerce

cf. [4]. It is important to make a distinction between to have *trust in* an object and to *act based on trust*. A specific online service can be perceived as trustworthy and a potential customer could do business with the actors involved if they decide to use the service and act based on trust. However, the service may not be needed for the moment but potential (future) users may still have trust in the actors providing the service.

Trust in online services isn't something that can be directly built into a computer system. Technology can however be used to engender trust [2]. Just as emotions such as happiness can't be directly expressed, signs and cues make it possible to assess why somebody or something is to be trusted [5]. Happiness may be expressed by signs such as a smile or in the way a person speaks. These are signs suggesting that somebody is happy. When cues suggest that somebody is trusted either by own observations or by means of others' recommendation we can use the notion *trustworthy*. To say that somebody is trustworthy is to say that based on subjective conclusions this person is to be trusted, a person has *trust in* another person or system.

1.1 Trust Definitions

Trust definitions can be found in literature within such disperse areas as biology, sociology, cognitive science, philosophy, and computer security but how the notion is used varies between the different areas. Some concepts mentioned in trust definitions are: *expectations, confidence, beliefs, subjective probability*, and *expected behavior*. In this section some trust definitions will be investigated in order to deal with RQ2.

One attempt to define trust was done by the psychologist Deutsch who did studies of the concept during the 60's and 70's [6]:

> "the confidence that one will find what is desired from another, rather than what is feared"

Another definition from the area of psychology was coined by Rempel et al. [7]:

> "a generalised expectation related to the subjective probability an individual assigns to the occurrence of some set of future events"

Another interesting view on trust is proposed by the philosopher Baier where trust is highlighted as something dependent on another persons will:

> "Accepted vulnerability to another's possible but not expected ill will (or lack off good will) toward one." [8]

The reasoning behind the definition is further explained:

> "One leaves others an opportunity to harm one when one trusts, and also shows one's confidence that they will not take it. Reasonable trust will require good ground for such confidence in other's good will, or at least the absence of good grounds for expecting their ill will or indifference."

This illustrates the view that trust can be about moral and ethical dimensions of human relationships. Gambetta present in the anthology Trust : Making and Breaking Cooperative Relations the following definition [1]:

> ”[Trust] (or symmetrically, distrust) [is] a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action. When we say we trust someone or that someone is trustworthy, we implicitly mean that the probability that he will perform an action that is beneficial or at least not detrimental to us is high enough for us to consider engaging in some form of cooperation with him. Correspondingly, when we say that someone is untrustworthy, we imply that that probability is low enough for us to refrain from doing so.”

In more recent work by Gambetta together with Bacharach [5] trust is defined as ”a particular belief, which arises in games with a certain payoff structure”. We will investigate the theories in their work in section 1.3.

The sociologist Giddens sees trust as:

> ”[the] confidence in the reliability of a person or system, regarding a given set of outcomes or events, where that confidence expresses faith in the probity or love of another, or in the correctness of abstract principles.” [9]

Giddens has a broader view on what we can place trust in namely, systems. From the perspective of online services this is interesting since most services involves a complex system of actors, agents and artifacts. Since Giddens is a sociologist it is important to point out that there is no 1-to-1 mapping of notions such as systems between sociology and computer science. Giddens theories will be discussed more in detail in section1.2 - *Trust and Expert Systems*.

The Trusted Computing Group(TCG) is an organization formed to develop, define, and promote open standards for hardware-enabled trusted computing and security technologies. This includes hardware building blocks and software. They also define important concepts in order to envisage trusted computing. Trust is defined as:

> ”the expectation that a device will behave in a particular manner for a specific purpose.” [10]

Implying trust to be about the behavior of devices. The focus of the Trusted Computing Group behind this definition is on hardware and software components. We will look closer at this in section 1.5.

1.2 Trust and Expert Systems

In order to better understand an interconnected trust framework and tools for an informed trust design (RQ2 and RQ3) Giddens theories on expert systems will be introduced. The system notion Giddens uses can be discussed in relation to online-services in order to better understand views on trust and systems of expertise. In Giddens work called *The Consequences of Modernity* [9] trust is claimed to have a major importance for many of the features that are typical for modern societies. Trust isn't just about trust in human beings as a whole. As a fact Giddens claims that in traditional societies this

was often the case but in modern societies, that is the western industrialized world, trust in other kinds of entities, *systems* is more often the case. Being more specific on the view of systems in Giddens work he uses the notion *expert system*. These are ”technical landmarks and professional expertise that organizes segments of the material and social environment we live in” [9]. We are constantly faced with, and surrounded by expert systems. When driving a car for instance we by some means rely on expertise that we hope has done everything to minimize the risks of getting injured. Traffic signs and rules are in place supporting this intention based on expertise about traffic safety and regulations. Typically drivers are aware of the involved dangers with driving a car but rely on expertise to minimize the involved risks as far as possible [9]. Giddens uses expert systems in a way that resembles expert systems as it is used in the Artificial Intelligence field in the sense that there is a body of professional expertise inside an expert system. Giddens sociological expert system is more about structures in society whereas expert systems in computer science typically are created to support decision making and knowledge preservation.

In the context of expert systems Giddens makes a distinction between the expertise of the systems (or representatives for them) and lay men. For lay men the trust connected to an expert system may depend on how the expertise or representatives behave. At points in time when laymen are interacting with the expert system this is done at *access-points*. The behavior of actors at the access points is important for how lay actors perceives the expert system from a trust perspective. Giddens example of a calmer flight attendant could affect trust through signaling ’business as usual’ towards lay man actors. In general the trust is in the expert system and not in the representatives or operators of it.

When talking of trust in expert systems Giddens uses the notion faceless commitments to describe what we put trust in meaning that we are trusting the system and its constituents, e.g. the body of knowledge, rules, codes of conduct etc. rather than its representatives. But Giddens state that their behavior may affect how well we perceive them to be regarding their competence to perform work within the expert system. When meeting a person face-to-face one can talk about facework commitments. Giddens doesn’t explicitly speak about technology as an access point toward technical solutions such as online services. Other scholars however, have suggested the Graphical User Interface of computer applications to be an access point where information is exchanged between lay men and representatives of expert systems [11].

Using access points in the context of computer systems like Arion et al. [11] proposes we see as interesting since it gives us a starting point to discuss trust in relation to complex systems involving technological components, social relationships and expert knowledge. It is important to point out that we are aware of the broader scope of Giddens theories regarding the implications for modernity.

1.3 Trust-warranting properties

Gambetta’s and Bacharach’s work in [5] brings up a number of concepts that are interesting from the perspective of trust and online services. All factors that govern trust between a truster and trustee can be referred to as trust warranting properties which are properties sufficing to make the trustee trustworthy from the perspective of the truster.

This can be both combinations of properties as well as a single property. The truster will not be able to observe the relevant trust properties by observation. Instead of observing *honesty* as a property the truster observes *signs of* honesty.

Gambetta and Bacharach make a distinction between unobservable trust warranting properties of a person (t-krypta) and observable features (manifesta). A passport in the immigration desk carried by an immigrant may exemplify observable features [5]. In the street the handbag of a woman is a manifestum. Depending on the media we are not able to observe the same manifestums of a person at all points of time. Using the telephone reveals the voice of a person but not the face and a letter reveals the handwriting of a person. Manifesta may be evidence relevant to signs of the t-krypta. Hesitation may be a sign of dishonesty but may also be perceived as a sign for that person's krypta for any other reason such as that the person is in general nervous [5].

We think the idea that signs mediates trust warranting properties is interesting when explaining the relationship between trusting and properties for trusting and how these trust warranting properties are perceived by an observer, in our wording a stakeholder. The ideas of trust in the light of manifesta, t-krypta and signs give us a vocabulary to further elaborate on trust in online service contexts.

1.4 Trust and Dynamics

Trust in stakeholders, artifacts and systems may change over time. A person may for instance trust another person to service a car and the expectations where surpassed. Thus trust in other humans can grow [7]. However, he or she may also be faced with disappointment and decide to rather let somebody else take care of my car reparations in the future. Trust can also decrease [12] due to disappointments. Since trust is dynamic it is very likely that there may need to adjust an online-service to encompass expectations of involved stakeholders. One area where concerns has changed over time is the online banking sector.

1.5 Other Trust Studies in Computer Science

Below we introduce a number of areas where trust is discussed in computer science. It is important to point out that these are just some areas where the notion is used.

Trusted Computing The trusted computing group's definition introduced in section 1.1 presented a view where trust is expectations of the behavior of a device. This definition and the work by the trusted computing group is mainly focusing on security aspects. The white paper *Trusted Computing is Real and it's Here* [13] presented on the TCG website presents the trusted computing (TC) area:

> "Trusted Computing is not just about key generation and storage. It's a whole host of related technologies being standardized at this very moment, including devices like phones, drives, networks, and servers; and various software[.....] Trusted Computing covers the whole spectrum of security technologies from a comprehensive, technically rigorous, vendor-neutral perspective."

This implicates that *trusted computing* refers to security mechanisms implemented by means of software. Hardware components are also part of the trusting computer domain since hardware adaptations often are necessary. One such example is *certified hardware configurations* where hardware is used to generate digital certificates to assure the claimed configuration [14].

From the perspective of this thesis trusted computing and similar approaches can function as tools to accomplish trustworthiness towards online services, but discussions about trust by means of a set of technologies won't give us a holistic view on trust in online services.

Trust and Security Often trust is stated to be identical with security. Trusted Computing above is one example and the concept of *Trusted Third Party (TTP)* another. This solution is used to establish a relationship between two parties on the Internet where the parties don't know each others' identity. In e-commerce services merchants and customers are involved in online transactions. The two may not know that much about the other entity hence there is a possibility that one of the actors in the transaction may misbehave [15] e.g. the customer may not trust the merchant to be the one he claims. In this case a trusted third party could provide credentials regarding the identity of one or more of the involved parties. Typically this is done by issuing an electronic certificate signed by the third party. This certificate is provided on the web site of e-commerce site. When the page is requested the certificate will guarantee the customer when mapped against the certificate authority that the identity of the web site is in line with the data on file at the trusted third party.

Trust and Multi Agent Systems In the Multi Agent Systems (MAS) world the notion of trust is used to mimic the role of human trust between software agents (agents will be used throughout this section exclusively to denote software agents). Work by Falcone et al. [16] was using the trust notion to reason about the probability that another agent will carry out a specific task (or set of tasks). Thus trust in delegation is the concept investigated in order to make sure that the probability that certain tasks are carried out as anticipated.

In order to enable trust in multi-agent systems discussions has been carried out in order to find mechanisms that enables the creation of agent societies where behavior and enforcements can be regulated in order to allow agents to interact more freely but not without potential sanctions. Here trust is seen as an important quality in order to decide who to interact with by means of task delegation. In the work by Falcone et al. [16] three categories of trustiers (trusters) was identified and their performance compared:

- *random trustiers* – are cases where trusters randomly chooses who to delegate tasks to.
- *statistical trustiers* – base their delegation of tasks to other agents on their previous performances
- *cognitive trustiers* – base their actions on socio-cognitive models that involves Fuzzy Cognitive Maps where specific features of the agents to place trust in (trustees) are considered. The features investigated are connected both to the trustee and the

environment. Willingness and abilities of the trustees are part of the map and these are connected to the specific task and the environment where the task is to be performed.

These three trusting mechanisms was implemented and tested and it was found that the cognitive trustiers where more successful that other agents. The role trust has in the MAS example here is as a role in societies of agents as a mechanism to deal with choices among different resources when it comes to task allocation and delegation. This is thus artificial agents *trust in* each other. We are not aware of if this is beneficial in MAS compared to other approaches where trust isn't used as a template. However, the research on agent-agent trust above imply that there may be reasons to use the trust metaphor in MAS contexts. There are certainly reasons to believe that a MAS that deliver online services based on proper implemented mechanisms can contribute to a positive trust experience.

1.6 Trust Attribution and Dimensions of Trust

In order to work with trust in the context of online services we have created framework where objects of trust (what can we have trust in) and trust dimensions (in regards to what we have trust in) has been condensed. The work in this section is based on the work mainly in paper 2: *Why trust is hard: Challenges in e-mediated services*.

Objects of Trust The studies of different theories of trust and trust definitions has lead to the following objects as starting points regarding who and what we can talk about *trust in*:

1. *Human trust and Confidence in social/natural order* – Our society rests on basic assumptions about what will and will not happen in most situations. For instance we have trust in natural order, that the heaven won't fall down or that the natural laws will cease to be true [17]. There is also a general trust related to the social order in most of our societies, that is that the governmental representatives will do their best for the citizens and countries they represent and follow laws and norms as well as follow established practices accordingly. This mutual trust isn't something that actors in general reflect consciously about. The non-reflective trust serves as a basic trust/confidence level for our daily actions where in general there aren't any alternatives to the anticipated risks. The notion *confidence* [18] is sometimes used in situations where actors in reality have no choice. It isn't a viable option to stay in bed all day due to concerns about the social or natural order. In the context of online services or use of computer systems and services to perform tasks there are underlying expectations serving for how we expect the services to perform. For instance the general perception of the Internet in most western countries today is that the infrastructure is stable due to proper supply of power and relatively stable infrastructure solutions. We are also expecting data to pass through the internet without governmental censorship etc. These preconditions has nothing to do with each and every instance of online-services used via the Internet. Instead these assumptions and expectations underlies all our expectations about online services.

2. *Human trust in humans* – In many situations we attribute trust toward other humans; we may trust a particular person about his capabilities or trust his intentions about a particular action. When buying a used car for instance we may trust a car salesman to a certain degree or trust a neighbor being an honest person. We may also trust our neighbor to look after our house or that the person approaching us on street asking for a donation. Trust between humans has been studied among others by [6, 19, 7].
3. *Human trust in artifacts* – The trusting relationship of type 3 is belonging to the group of human made objects such as cars, computers, VCR:s are in some cases discussed in a manner which implies that these objects can be seen as objects in which trust is placed. For instance 'I trust my car' or 'they trusted the bus to arrive on time'. This means that our expectations regarding the objects with respect to reliability are in some sense confused with or attributed for trust in humans enabling the intended behavior (the design and implementation team of a company, the driver of the bus employed by a public transport company). Since it is unusual or questionable to discuss classical artifacts as trustworthy entities with bad will (or good) toward others or as in possess of emotions the use of the notion trust in artifacts is not classically applicable in those settings. However, an artifact can be designed or equipped in a reliable manner with good functionality. Artifacts also can mediate and present suggestions that the involved actors are trustworthy. However, one may probably use the term *reliability* [2]
4. *Human trust in communities and organizations* – Humans are often part of a larger community. In the society we have for instance companies, non-profit organizations, governmental institutions and other groups of humans, which often act according to policies, and interests of the community. In many cases the trust may be attributed primarily (or at least in part) in the behavior in a community e.g. a hospital. On the other hand, a hospital may be perceived as more trustworthy than another due to better reputation regarding the perceived treatment and quality of their staff. Depending on the context, trust by a subject may be placed on the object being a community, an individual representing the community, or both. This kind of trusting object we see in work by Barber [17] and Giddens [9].

Dimensions of trust Trust concerns are related to one or more trust dimensions that relates to generic qualities that can be associated with the object of trust:

1. *Trust in Professional Competence* – When a decision to delegate a task to another actor is taken this is often based on a perception of that actor's professional competence. This refers to expectations about the professional abilities [17] of e.g. a doctor or banker and suggests further refinements of trust expectations. We can trust somebody to have the right competence for carrying out actions associated with their profession. We trust doctors' judgments about medical needs and we trust them in their ability to adjust treatments in accordance with new findings within their area of expertise.
2. *Trust in Ethical/moral Behavior* – Trust isn't only related to professionalism in dealing with tasks as such, it is also suggested to be linked to values and less tangible nuances such as ethical and moral premises. If a trusted professional acts in

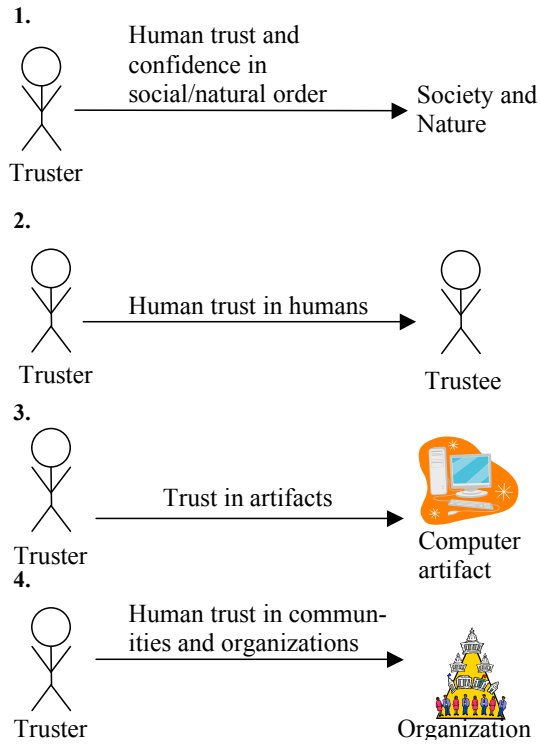


Fig. 1. Four categories illustrating how trust can be discussed in the context of online services

a manner that is perceived as being against common ethical and moral norms we can choose to distrust this person in a given context despite his professional skills. Examples include certain types of medical experiments or other acts that can be regarded as unethical or even criminal if detected. Trust in moral or ethical behavior is, of course, very context dependent. Moral trust or as it is put forward by Barber as trust in fiduciary obligations [17] means that some others in our social relationships have moral obligations and responsibility to demonstrate a special concern for other's interests above their own. The lack of control will give the trustee the possibility to exploit or harm the truster [8]. The ethical/moral trust dimension is based on a scenario when there is a risk for betrayal based on ethical and moral reasons. For instance in an online service the information handled by the involved organization about individual clients can be misused in unethical manners in a way that is perceived as unmoral and would harm the truster. This is also connected to willingness from the trustee to put the truster's interest before his or her own.

3. *Trust in Action Fulfillment* – In cooperation a specific trust dimension surfaces in most contexts. That is, can a subject trust that an object will indeed fulfill a promise or obligation to do a specified action? In a subcontractor scenario or in a health care

situation where a doctor has prescribed a treatment to be carried out concerns may occur about whether the treatment will be carried out or not. Similar concerns can be identified in online services regarding whether e.g. an ordered product will be delivered or not.

4. *Functionality* – The functionality of an artifact is an important and natural quality of trust, e.g., the tools are expected to function as they should. An implicit trust condition is that an artifact or tool is not behaving in an unexpected or undesired way by its design [20]. As we have indicated earlier, this situation is quite different when it comes to computer (software) based artifacts, that is, online services. Firstly, the available functionalities, or affordances, are more complex (flexible). Secondly, and more important from a trust perspective the software can be designed by purpose or by affording vulnerabilities to create dysfunctional behavior that can be very harmful to the user or her system. The explicitly available functions and their appearance and accessibility shape the online service from the perspective of its users. The user has to trust that these services meet her trust criteria in a trustworthy way without unwanted results.
5. *Reliability* – The reliability of an artifact is another important criterion of trust in artifacts. The tools should be resistant to tear and wear in a reasonable way and the e.g. a VCR should function flawless for some years. Reliability thus means that an artifact can be expected to function according to the presented functionality and is working when needed. [2, 20]

The views in this section describe our view on trust that will be used throughout the thesis. It is also important to point out that the above-mentioned categorizations have developed during the writing of the included papers and that there are discrepancies of concepts throughout the thesis. In this theoretical background we use the most recent definitions.

2 Designing Online Services

Building online services/e-services and understanding needs and concerns of involved stakeholders is a complex task. An e-service can be defined as an ”Interactive software based information systems received via the Internet” [21]. We would however like to emphasize that there may be reasons to define other Interactive software based systems as e-Services, for instance an information system for providing mobile users with information from another mobile device. The difference may be means of distribution and an open network such as the Internet poses certain security concerns compared to more limited systems. When we use the notion of online services we have this broader setting in mind[1]. We perceive online services as a *software intensive system (SIS)*. A SIS is a system where the software essentially influences the design, construction, deployment and evolution of the system as a whole [22]. Thus the *system* in our case is the online service. The IEEE defined SIS in the *IEEE Recommended Practice for Architectural Description of Software-Intensive Systems* [22]. SIS is interesting since the use of the

[1] Throughout the thesis both e-services and online services are used. We use the notions interchangeable

notion implies a more holistic view on software based systems, that is, these are embedded or part of larger systems that they can't be separated from. The term system used in the SIS standard defines a system to be a collection of components organized to accomplish a specific function or set of functions. In the context of online-services hardware components e.g. computer devices used to create the online service.

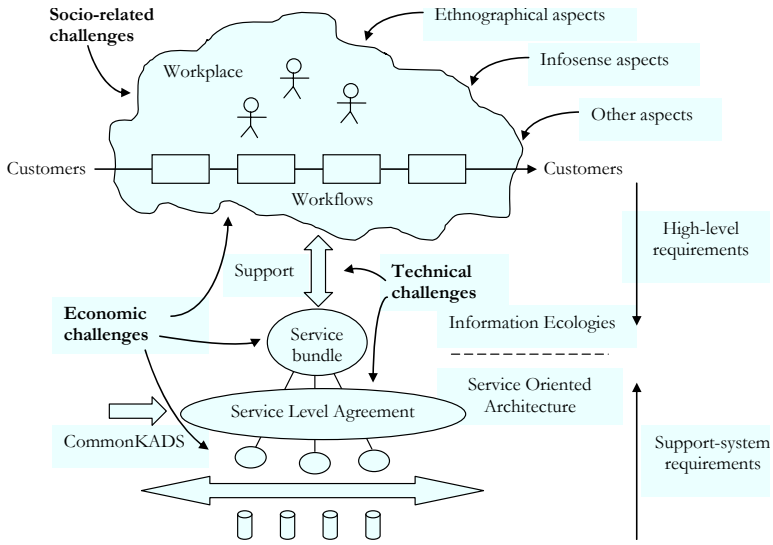


Fig. 2. A socio-economic-technical system with online service composition for workflow support. From [23] and [24].

A number of aspects related to that context are depicted in fig. 2; ethnographical aspects, giving insights into how and what people do, and Infosense (what kind of data and information that is exchanged) aspects are depicted. These are two approaches that can be used to better understand the needs on which supporting tools e.g. online services should be based on. From the perspective of socio-economic-technical systems the tools presented in this thesis are intended to give insights into aspects of a context related to trust and trustworthiness of tools used in the workplace context. The model presented in fig.2 depicts components of the socio-economical-technical system where the focus is on work-flows at a workplace. A work-flow is a set of interconnected sequential tasks [23, 24]. At the workplace there are a number of stakeholders who are carrying out the tasks in a number of work-flows. These can be supported by the provision and use of one or more (online)-services. These are created based on service level agreements specifying rules for the composition. In order to develop online services the model developed by Östlund and Brandt [24, 23] can serve as a starting point in order to understand the complexity of both contexts and challenges. Three challenge areas has been identified:

- Socio-related challenges – mainly connected to the work-flows and involved humans.
- Economic challenges – includes efficient pricing and cost levels both in the workplace context, and the composition and pricing of services.
- Technical challenges – how can the technology be developed and adjusted in order to deal with economically sustainable and profitable online-services and efficient work-flows in the workplace context?

The challenges also affect how the technology e.g. online service software should be designed; the support arrow connects the workplace context and the service bundle that constitutes of software based services. These software services needs to be built properly in order to form stable software that can be modified and adjusted according to technical, economical and sociological needs.

2.1 System Engineering

System engineering is an activity concerned with the specification, design, implementation, validation, deployment and maintenance of systems. A system can be defined as: "a purposeful collection of interrelated components that work together to achieve some objective" [25]. A system can in this sense consist of hardware, software and human users [25]. An online service is one example of a system that in turn may consist of other systems for distribution, hardware systems and software systems that are interrelated. A system is located both in a physical and organizational environment which affects how the system should be designed. This is due to the fact that most systems are designed to affect its environment. Furthermore the environment may change over time affecting how well the system will function[2] [25]. In order to design a system and in order to identify requirements de-composition takes place. A system can typically be decomposed into a number of subsystems which in turn can be further de-composed. The lowest de-composition level is into functional components. This level is reached when the component from the perspective of the subsystem provide one distinct function. There are a number of distinct differences between Software engineering and System engineering [25]:

- *Interdisciplinary involvement* – engineers from different disciplines are involved opening up for misunderstandings due to terminology mismatch.
- *Less flexible system development* – in complex systems involving e.g. antennas or material structures changes can be almost impossible to do when a decision has been made and the implementation started. Software is an exception since it can be adjusted to reflect new or changed requirements during a larger portion of the system development process.

The system development process consist of the following stages [25]:

- *Requirements Definition* – here the requirements for the system is specified.

[2] Sommerville seem to focus on factors that affects the systems stability and ability to address its objectives by means of functions rather than on purposeful functionality *in* the environment as such.

- *System Design* – the design of the system takes place. This typically includes decomposition into sub-systems and their interfaces. The sub-systems should also be assigned with relevant requirements from the previous stage.
- *Sub-System Development* – the sub-systems are developed.
- *System Integration* – the sub-systems are integrated into the desired system.
- *System Installation* – the system is installed where it is intended to operate.
- *System Evolution* – the system may need to be changed due to e.g. changes in the environment or new requirements.
- *System Decommissioning* – a system will sooner or later be taken out of operation. This may require large efforts as in the case with nuclear power-plants. Often software may have lesser implications when it comes to e.g. environmental aspects.

The system development process inspired the first software development processes such as the waterfall model [25]. From there the engineering practices has been refined to better suite the engineering challenges associated with software.

2.2 Software Engineering

In order to foster the implementation of software a good and structured approach is helpful in order to accomplish the complex task of software construction. The need to formulate a field was discovered in the 1960s when a series of NATO conferences where organized on the topic[3]. Since then theories, methods, and models has been developed and improved. The need for good approaches to software development still pertains but the complexity with more complex business models, networked computing and the fast paced technical development still makes each and every software development project to a true challenge. *Software engineering* (SE) can be defined as:

> ”an engineering discipline which is concerned with all aspects of software production” [25]

Software engineering doesn't just focus on technical issues, but also on project management, tool development and software production theories and methods. Thus the area has a rather broad scope that touches many important aspects of software development. When working with the development of software a software process is used. A software process commonly consist of [25]:

- a Software specification where the functionality and constraints of software is defined.
- The software development tasks that are carried out in order to meet the specification.
- The validation task where software is validated, in order to motivate that it does what the customers want it to do.
- Software evolution, that is there is typically a need to redesign and change portions of the solution to encompass changing customer needs.

[3] For more information about the NATO conferences on software engineering visit: http://homepages.cs.ncl.ac.uk/brian.randell/NATO/ (accessed 2007-05-24)

Trust issues that we claim need attention thus must be able to be addressed in all of the software process components without the risk to be misrepresented. Good models for software development ideally enables us to deal with and reason about trust issues in a sound manner.

2.3 Requirements Engineering

Within the software engineering discipline there are areas that has received specific attention such as software architecture that deals with how to structure software products in order to increase flexibility and improve maintenance. Another area is requirement engineering that is concerned with the process to understand and elicit requirements that can be translated into software [26]. Programming languages are formal descriptions that are executed in a binary format thus the requirements must be reflected in a manner that enables consideration in a future implementation.

Requirements engineering provide a toolbox for the process of elicit, handle and assess requirements. Moreover developers can verify and validate their efforts. The process of verification means to carry out a check, where it is investigated if the developed system satisfies the requirements. The process of validation is the process where the customer must be able to validate the set up requirements to see that the needs are met. As Lausen puts it the customer must be able to understand the specification and say: ”Yes, this is what I need- This system will solve my problems” [26]. Requirements are based on identified demands which needs to be turned into requirements by an analyst, requirements engineer or similar.

The process to identify requirements takes place early in the development of software. The requirements are formulated and documented in a requirement specification [26, 27]. This is often an important part of the agreement between the customer and the developer, so it is both a document used in the rest of the development of the software as well as a legal document. The requirements should be based on demands related to one or more stakeholders. Different users or user groups, the customer’s IT-department and in some cases also external parties are all examples of potential stakeholders [26]. The person responsible for the process of understanding needs has a challenging task since s/he needs to enable the stakeholders to express their demands in a clear way and understand problems that arises when there are conflicting demands. Also stakeholders may have trouble to express what they want or need or have trouble understanding the development process. Last but not least, demands may change over times due to other functions in other systems, changes in regulations, or changes in soft- and hardware environments. Thus the complexity in the field of requirements make the process to elicit and specify the perfect set of requirements difficult. Ideally all inputs should be specified and in conjunction with the input a corresponding output [26]. However, software intensive systems are complex and to reflect the input-output sequences in detail for all possible uses is a task that can hardly be accomplished. Instead requirement engineers tend to used tools that reduces the complexity of the requirement specification work. By specifying functional requirements e.g. “when button b is pressed the screen should turn black” is one way to reduce the complexity. This may also affect the precision but will in many cases be good enough. According to Lauesen [26] the most common requirements categories is one of the following:

- *Data Requirements* – what input and output should be handled and what kind of data should be stored in the system?
- *What kind functionality should be implemented into the system?* – e.g. should there be a print function or not?
- *Non Functional, or Quality Requirements* - What kind of qualities should a system have? Performance and usability are examples of non-functional requirements.

Except for these there are managerial requirements as well as economical that has impact on the design of software. Requirements should be traceable both forward and backwards [26, 27]. The backward tracing goes from requirements to demands, to justify the presence of a requirement. Another backward tracing portion has to do with the fact is to see if all portions of the program are required. This is done in order to avoid that parts are implemented that the customer didn't ask for. Forward tracing can also be divided into two parts. The first is concerned with the process to see that all requirements specified are dealt with in the solution. The other is to inspect that all demands are reflected in the requirements [26].

Requirements engineering is a process essential for systems designers and programmers in order to know what they should build. The verification process and requirements document serves as a basis for formal contracts and discussion between the customer and the developer of the system. This means that in cases of conflict the requirement specification can give the conflicting partners guidance on what has been agreed upon.

3 Activity Theory

One area that has been in the background during the work with this thesis but which we think can contribute to a better understanding of online service contexts and their use among humans is Activity theory. This is an area with roots in psychological Russian research featuring a non-reductionalistic approach on understanding of human activity [28]. Contextual factors such as artifacts and context are often considered as important as internal cognitive processes when understanding the human mind. The unit of analysis, *the activity* is motivated by a set of stable and sustainable relations between the included concepts. One of the claims made by the activity theory community is that the nature of an artifact only can be understood within the context of human activity by identifying the ways people use this artifact, the needs it serves and its history of development [29]. From the perspective of trust in online services it suggests a necessity to understand the contexts where the online services are part of an activity. For a nurse, the online service for trusted delegation of tasks (introduced in the next section) is a tool in her nursing activity. In order for online service designers an understanding of how a proper tool for task delegation will work in the nursing activity a better understanding of the activity therefore could open up for new needs and potentially new products. We will give a short introduction of the trusted task delegation tool here. For more information see appendix A on page 115.

3.1 Home Healthcare Group-Planning Scenario

In order to carry out work in the healthcare domain in Sweden tasks can be delegated by doctors to nurses and nursing assistants. For some of these tasks the doctor needs

to know that the nurse is capable of carrying out the task. If something goes wrong the doctor who ordered the task to be carried out based on a delegation is responsible for the consequences if the task is not within the formal work description of the nurse or nursing assistant. Thus if a doctor has trust in the professional competence of a nurse or nursing assistant he can award that person with a certificate that s/he is qualified for carrying out such tasks. This could be to give medication or injections. The purpose of the trusted delegation tool is to illustrate how a distributed solution could look like that assists the delegation of tasks based on issued certificates across a distributed health care organization such as in home health care settings.

Role	**Role related skills**	**Skills that can be certified**
Doctor Has power to issue certificates for all skills a nurse or an assistant nurse can perform with a certificate. Has power to announce tasks requiring any role related or certifiable skill of a nurse or assistant nurse Have power to accept tasks requiring any of their role related skills. and any skill for which they can issue certificates	Issue prescriptions Perform a diagnosis Also has all the skills role related and certifiable skills of nurses and assistant nurses	
Nurse Has power to issue certificates for all skills an assistant nurse can perform with a certificate, except insulin injections Has power to accept tasks requiring skills that are included in their role related skills, or the skills for which they can issue certificates, or the skills for which they have a certificate	Give morphine injections Give medication according to prescription Give some medications without a prescription All role related and certifiable skills of assistant nurses	Issue prescriptions of pain killers (requires certification for skill: issue prescriptions of pain killers)
Nursing assistant Has power to accept tasks requiring skills that are included in their role related skills, or the skills for which they have a certificate	Change bandaging on sores Read and report body temperature	Give certain prescribed medications to patients Place catheter on patient Give insulin injections

Fig. 3. A table of role related- and certifiable skills for an online service for trusted delegations of tasks.

A particular agent in 3 with an associated role not only can carry out tasks related to that role but also delegate a task to a particular agent with another role as indicated in the table. However, the possibilities to do so are based on a certification process. The certification process involved gives certain agents the right to carry out certain task

during a limited time period. These certificates can be withdrawn and updated for new time periods. Delegation of tasks is both regulated by laws and regulations as well as by internal rules and procedures within the care context. The solution is implemented to be used on mobile devices and are connected to a Prolog rule base that formally specifies roles and responsibilities. When a certificate is awarded or withdrawn for a particular agent this is immediately reflected in the rule base. This means that awarded tasks will be revoked if a particular nurse don't have any valid certificate when the task is to be carried out.

3.2 Modeling activities

One contribution to the field of activity theory that opens up for the possibility to reason and model about complex relations of human activity has been made by Yrjö Engeström [28]. The work resulted among many contributions in a framework and notation for human activity. The framework is in our opinion an interesting tool to create more informed online service designs where qualities of trust issues can be determined not only within the service itself but in contexts where the service is used as a tool. The components of Engeströms framework are:

- Subjects. The subject of an activity is the human person involved in an activity operating on the object in the activity. For instance in the activity *home health care group planning* the subject could be the doctor responsible for the workgroup planning.
- Objects, the object is the focus for the activity. The actions and operations carried out are all focused towards the object. This is the planning material.
- Outcome of the activity. This specifies the outcome of the activity, e.g. from the planning scenario a plan that specifies who should do what task.
- Division of Labor. Specifies both the formal structures such as organization as well as the informal division within an activity.
- Community. The subjects part of and related to the activity in question. An activity may have many different subjects involved with different roles.
- Tools, which includes both physical tools such as hammers, computer, schedules, models and other man made constructs such as language.
- Rules, both formal rules i.e. laws and regulations as well as informal rules and practices are defined here.

The components are described in relation to each other in fig.4.

An interesting feature in activity theory is that it emphasizes the idea that interaction with computer systems or online services in itself isn't the goal for the users. It's the goals associated with the context where the artifact (in this case the computer device with associated software) is used. In the example above for example the activity the nurse is carrying out isn't primarily concerned with the use of the computer system or even the results of the online-service. Instead the primarily focus is the object of interest in the activity, i.e. the care receiver. An activity also have a history. Rules may be based on old tools and division of labor may also be based on older tools.

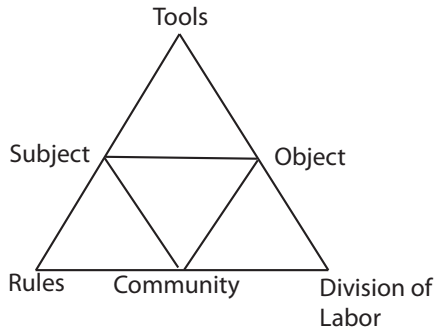


Fig. 4. The Activity Theory Model proposed by Engeström [28]

3.3 The Activity hierarchy

An activity involves a number of steps in order to manipulate an object. In home health care group planning for instance a doctor may issue warrants, send messages to a nurse, type on a computer etc. In activity theory tasks can be classified into the following hierarchy described top-down as follows:

- *Activity* - The unit of analysis as described above and the motivation for the actions and operations directed towards an object. In the case above, the home health care group planning is the activity.
- *Actions* - are conscious steps taken with an explicit goal. In the home health care group planning activity this may involve accessing the online service to provide instructions. Another action is to issue a warrant for a nurse or nursing assistant.
- *Operations* - on the other hand are subconscious steps taken within the activity. For instance, typing using a keyboard or mobile device.

The levels of human activity above are by no means static or absolute; an action can become an operation as well as an operation can become an action. For instance if a problem occurs during typing, e.g. a malfunctioning key or input command, typing operation becomes a conscious step. Typically new practitioners in an activity may have more actions in an activity, but as they learn and becomes more familiar with the practice these actions becomes subconscious steps in the activity i.e. operations. All activities are per definition collective. This doesn't mean that subjects need to cooperate explicitly with other humans, but it means that our actions and operations directed toward objects are always related to, and affect other human beings. For instance when writing text in a word processor the text is in some sense always intended to be read by other human beings [30]. The triangular structure of the activity denotes relationships between the components of the model and relates to ideas discussed within the area of activity theory e.g. mediation. In activity theory the technology is clearly subordinated humans (subjects); humans are not just another component in a system [29].

3.4 Mediation

Technology is a tool that mediates manipulation of an object. In order for us to manipulate the object we use tools [30, 28]. When hammering a hammer is used or another convenient tool that can be used in order to hammer. Online services are used as a tool in some kind of activity, e.g. for home health care group planning. A set of tools (e.g. computer, mouse, web browser, online-service functionality, pen and paper, medical vocabulary) to manipulate the care receiver. No direct manipulation of the object by the subject is possible without the use of tools [28].

3.5 The history of activities

The history of an activity system is important in order to form an understanding of why an activity is constituted the way it is. One example, the activity of chopping down trees carried out by lumberjacks is an ancient activity where rules, community, division of labor and the look of tools have changed over time. The object, trees has been static together with the subjects of the activity (lumberjacks). Depending on rules, division of labor and the constitution of the community the axe have been developed over centuries into the shapes it has today. The development of the axe and other tools such as models and languages has been created and reshaped over time depending on problems discovered in its use and depending on the activities used to create and manufacture the tools. The use of the axe and its history of development are embedded in the current axes which most cases are improvements and adoptions of earlier versions or tradeoffs made between different actions within an activity. To fully understand an activity it must be placed in a context of other activities where the relationships to other activities can be understood.

3.6 Activity Systems

An activity system consists of related activities where the components is related to each other. An activity is the smallest meaningful unit of analysis when it comes to human activities [28]. However, activities are not an isolated phenomena, they are connected to other activities. All activities consume resources and produces some kind of outcome by manipulations of objects [28]. The tools used by online service designers constitutes of tools developed in another activity.

Consider for instance the activities involved in the care of a patient. A doctor makes diagnoses and treats patients. Nurses are involved in the process as well. Both have the same object namely the patient. In the treatment tools are developed in order to provide the nurses and doctors with tools for their work. Some activity systems are very stable. An example of this could be a hospital where the involved activities have been provided throughout the latest centuries. Others are less stable and may disappear or change its shapes very often.

4 Concluding Remarks

A number of areas has been discussed as background for the content of this thesis. Trust and trustworthiness has been discussed describing what and how it can be related

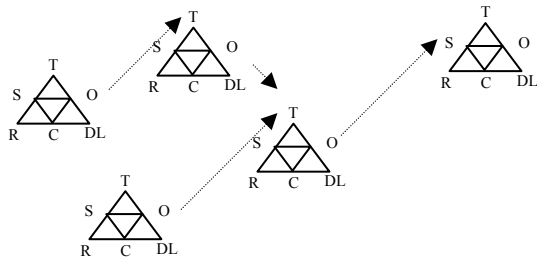


Fig. 5. An activity system consisting of a number of activities. In this example there are relationships between the activities where the object in one activity is used as tool in other activities. As illustrated there can be many-to-one relationships between activities. It is also possible to have many-to-many constellations as well in order to illustrate the connections between different activities [28].

to online services. There are many theories on what trust is and what function it has in our lives. Also what we can have trust in varies according to different scholars. The division of what we can have trust in and dimensions of trust was presented in section 1.6 - *Trust Attribution and Dimensions of Trust*.

We have introduced our view on online services and theories of software intensive systems where we see online services as software intensive systems. We have also discussed systems engineering and software engineering in brief and particularly requirement engineering. It is probably hard to specify the 'right' requirements from all stakeholders and balance them correctly. Furthermore requirements may in a system(that is not just the software system) be elicitated for different sub-components. It is a challenge to coordinate all these efforts in our view since interpretation and prioritization of these may affect the trustworthiness exhibited by online services.

The system engineering process give us insight into the design of (technical) systems in general. The step *System decommission* is often abandoned, at least explicitly in software development processes. This we think is a step that may prove to gain importance since many features are depending on software based components. Especially in online services where a service-oriented approach may be considered. These may depend on sub-services. How can a proper decommission of a sub-service be realized without the risk of problems with associated services? Here we may even need regulative measures in order to enable online service providers to deliver trustworthy online services.

To support a better understanding for the contexts where online services serve as tools for stakeholders Activity Theory can be used. This will ideally enable us to identify stakeholders' problems related to trust and trustworthiness better since we are able to reason about the use of the online services in socio-technical contexts.

References

1. Gambetta, D., ed.: Trust : making and breaking cooperative relations. B. Blackwell, New York, NY (1988)

2. Friedman, B., Kahn, P.H., Howe, D.C.: Trust online. Communications of the ACM **43** (2000) 34–40
3. Luhmann, N.: Trust and Power: two works. John Wiley, Chichester (1979)
4. Egger, F.N.: Affective design of e-commerce user interfaces: How to maximise perceived trustworthiness. In Helander, M., Green, W., Tham, M.P., eds.: Conference on Affective Human Factors Design, Singapore (2001) 317–324
5. Bacharach, M., Gambetta, D.: Trust as type detection. In Castelfranchi, C., Tan, Y.H., eds.: Trust and deception in virtual societies. Kluwer Academic Publishers, North Holland (2001)
6. Deutsch, M.: The resolution of conflict; constructive and destructive processes. Yale University Press, New Haven,NY. (1973)
7. Rempel, J., Holmes, J., Zanna, M.: Trust in close relationships. Journal of Personality and Social Psychology **49** (1985) 95–112
8. Baier, A.: Trust and antitrust. Ethics **96** (1986) 231–260
9. Giddens, A.: The consequences of modernity. Polity Press ;, Cambridge (1990)
10. Trusted Computing Group: Glossary of terms. https://www.trustedcomputinggroup.org/groups/glossary/ (2007) Accessed 2007-08-28.
11. Arion, M., Numan, H.J., Pitariu, H., Jorna, R.: Placing trust in human-computer interaction. In Opperman, R., Bagnara, S., Benyon, D., eds.: Seventh European Conference on Cognitive Ergonomics (ECCE 7). GMD-Studien Nr. 233, Bonn, GMD (1994)
12. mcKnight, H.D., Cummings, L.L., Chervany, N.L.: Initial trust formation in new organizational relationships. Academy of Management Review **23** (1998) 473–490
13. Kay, R.L.: Trusted computing is real and it's here. White paper presented on trusted computing group website: https://www.trustedcomputinggroup.org, Endpoint Technologies (2007)
14. Felten, E.: Understanding trusted computing: will its benefits outweigh its drawbacks? Security & Privacy Magazine, IEEE **1** (2003) 60–62
15. Gollmann, D.: Computer Security. Wiley & Sons (2005)
16. Falcone, R., Pezzulo, G., Castelfranchi, C., Calvi, G.: Why a cognitive trustier performs better: Simulating trust-based contract nets. In: AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, Washington, DC, USA, IEEE Computer Society (2004) 1394–1395
17. Barber, B.: The logic and limits of trust. Rutgers University Press, New Brunswick, N.J. (1983)
18. Luhmann, N.: Familiarity, confidence, trust: Problems and alternatives. In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. Basil Blackwell, New York, NY (1988) 94–110
19. Gambetta, D.: Can we trust trust? In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. B. Blackwell, New York (1988)
20. Muir, B.M.: Trust in automation .1. theoretical issues in the study of trust and human intervention in automated systems. Ergonomics **37** (1994) 1905–1922
21. Featherman, M.S., Pavlou, P.A.: Predicting e-services adoption: a perceived risk facets perspective. International Journal of Human-Computer Studies **59** (2003) 451–474
22. IEEE-SA Standards Board: IEEE recommended practice for architectural description of software-intensive systems. Standard Description 1471-2000, IEEE (2000)
23. Östlund, L.: Information in use: In- and Outsourcing Aspects of Digital Services. PhD thesis, Blekinge Institute of Technaology (2007) ISSN 1653-2090 ISBN 978-91-7295-110-5.
24. Brandt, P.: Information in use: Aspects of Information Quality in Workflows. PhD thesis, Blekinge Institute of Technology (2007) ISSN 1653-2090 ISBN 978-91-7295-111-2.
25. Sommerville, I.: Software Engineering, engl. Ed. (International Computer Science Series). Addison-Wesley Longman, Amsterdam (2000)
26. Lauesen, S.: Software requirements: styles and techniques. Addison-Wesley, Harlow (2002)

27. Kotonya, G., Sommerville, I.: Requirements engineering : processes and techniques. Worldwide series in computer science. J. Wiley, Chichester ; New York (1998)
28. Engeström, Y.: Learning by expanding : an activity-theoretical approach to developmental research. Orienta-Konsultit Oy, Helsinki (1987)
29. Kaptelinin, V.: Computer-mediated activity: Functional organs in social and developmental contexts. In Nardi, B.A., ed.: Context and Consiusness. MIT press, Cambridge, MA (1996) 45–68
30. Bødker, S.: Through the interface : a human activity approach to user interface design. L. Erlbaum, Hillsdale, N.J. (1990)

Part II

Included Papers

Paper 1: Computational ecosystems in home healthcare

Gustavsson, R., M. Fredriksson and C. Rindebäck (2001)

Appears in:
Social Order in Multiagent Systems. C. Dellarocas and R. Conte. p.201–220.
Kluwer Academic Publishers. Boston

Computational Ecosystems in Home Health Care

Rune Gustavsson, Martin Fredriksson, and Christer Rindebäck

Department of Software Engineering and Computer Science
Blekinge Institute of Technology
S-37225 Ronneby, Sweden.
rgu@ipd.hk-r.se, mfe@ipd.hk-r.se, cri@ipd.hk-r.se

Abstract. The focus of this chapter is to evaluate how to appropriately apply information technology and computational ecosystems in electronic health care without sacrificing the quality of service. We conduct this evaluation by introducing two scenarios (SMART CARE and HOME DIALYSIS) and a trust enforcing model (ORA). Consequently, the system design process of trust enforcing ecosystems is also introduced and discussed. The approach described in this chapter aims at clarifying the need for institutions (as we perceive them in human societies) to be implemented as a fundamentally important part of computational ecosystems that are grounded in both the real world and a virtual environment.

Keywords: Computational ecosystems, electronic health care, trust, and institutions.

1 Introduction

Social interaction concepts such as norms, commitments, obligations, rights, permissions, responsibilities and so on have been studied from several points of view in the area of multiagent systems. Many of these studies have focused on different aspects of electronic commerce. However, recently a new and important domain for applied information technology has emerged - electronic health care. The focus of this application domain is how to use information technology in order to cope with an increasing demand of health care services and at the same time manage the increasing costs in the social and health care systems without sacrificing the quality of services. In Sweden, as elsewhere, the possibilities of information technology in electronic health care applications are evaluated in several national programs. Blekinge Institute of Technology is involved in two national projects related to these programmes: SMART CARE and HOME DIALYSIS.

A common goal of the projects is to identify and implement information technology support which allows aging citizens to live a secure, safe, and comfortable life in their homes as long as possible. Proper selection and gradual introduction of "smart home" devices and services on a pre-installed infrastructure [1, 2] is the work agenda of the projects. The HOME DIALYSIS project aims at transferring health care treatment, now

performed at hospitals, into properly equipped homes. A big challenge here is to re-assess routines at hospitals into a distributed set of new routines, while keeping (and if possible increase) the quality of the treatment. A key issue is how to gain and maintain trust, from all parties, in the new situation. In SMART CARE we are mainly focusing on new and emerging technologies supporting life-critical functions in smart homes built around the needs of elderly people or people with special needs. The technologies, or equipment, are developed and installed by our partners in the NATIONAL ADVANCED HOME HEALTHCARE ENVIRONMENT project of which SMART CARE is a part. We can foresee that in the near future we will even have smart implants supporting life-critical functions, "smart bodies".

The focus of the SMART CARE project is on smart homes built around the needs of elderly people or people with special needs. The HOME DIALYSIS project focuses on the possibilities of managing dialysis at home. Needless to say, in both these cases the acceptance and hence the usefulness of project results are due to aspects of a very non-technical nature. Issues such as responsibilities and trust are in the foreground for an acceptance of services required by patients. It is also inevitable that the introduction of information technology-based versions of these services will change how the health care providers operate and how their personnel are trained. In the SMART CARE project new information technology will enable the creation of a computational ecosystem around the care receiver. That is, the information created around and by the care receiver has to be gathered, processed and distributed in a way that makes life for the care receiver both safer and worthwhile. At the same time the "information profile" that can be generated from the smart body and the smart home can be very sensitive and has to be of high integrity to be trusted. In the HOME DIALYSIS project, tasks currently performed at hospitals are transformed into tasks performed at the premises of the patients. Obviously, issues such as responsibilities and trust have to be addressed and empowered by the technical solutions at hand, and at the same time we also have to address the role of culture as a fundament behind issues such as norms.

In the following sections of this chapter we will describe two different scenarios (home dialysis and smart care) in order to address the issue of trust in change. Our approach to an understanding of this issue is to apply a model comprising concepts such as ownership, responsibility, and accessibility (i.e. ORA) which we argue are fundamental cornerstones in the operationalization of trust. The implementation of a computational ecosystem, in a societal setting, is also discussed in terms of a software infrastructure and architecture. Finally, a summary of our main findings is presented as well as suggestions for future work.

2 Scenarios and Experiments

A fundamental issue in health care is trust. That is, a person has trust in another person to successfully perform a task in a given context. In this particular case trust is often related to a person acting in the context of institutional empowerment, i.e. a nurse in a hospital. Furthermore, trust is also often related to responsibility. In a hospital we trust the quality of treatment because we can see and accept a certain chain of responsibilities. However, a basic question: how can we develop and implement information

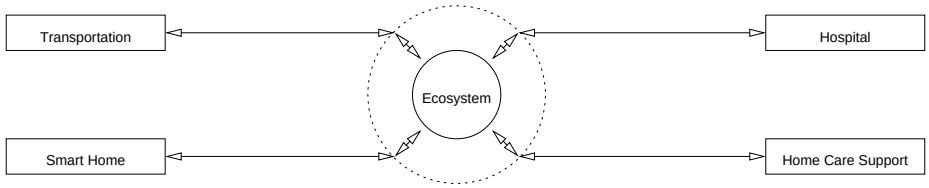


Fig. 1. Basic features of an electronic health care system: transportation, home care support, smart home, and hospital (cultural aspect of the system). The dotted area in this figure depicts an ORA mirror.

systems that allow us to have trust even in health care partly constituted by a supporting information system?

Figure 1 highlights important aspects of this type of health care systems. First of all, caretaking processes can take place at three different locations; at home, in hospitals or in between. In the latter case it could be in an ambulance or other transport or at another institution such as a local health care or activity centre. Secondly, the figure points out the importance of institutional culture associated with health care measures. Thirdly, the figure highlights the importance to ground the information system (e.g. an artificial world) in the real world.

Later on, in a design example of a computational ecosystem (see Section 6) we make use of entities in an *artificial world* to reflect both notions and behavior corresponding to smart equipment, services, and people in the *real world*. As we will discuss further on important connections between the real and the artificial worlds are provided by ownership, responsibility, and accessibility relations (see Section 3). A closer look at two electronic health care scenarios reveals a set of challenges to address and resolve before we can implement the new models of health care.

Home Dialysis Scenario Presently, dialysis sessions are mainly performed in hospitals. Patients have to visit the hospitals on a regular basis, several times per week, and spend many hours at the hospital each time. Needless to say, this is a costly and time-consuming form of treatment. Due to technological advancements there are now possibilities to allow patients to receive treatment at home. Different methods to electronically connect the patients' home with the hospital are tested at the moment. From a trust perspective there are, however, several open questions emerging from the present hospital-oriented work habits.

Firstly, patients that are treated in hospitals are by and large seen as passive objects undergoing treatment from the perspective of the hospital personnel. Secondly, in hospitals there is a responsibility chain based on locality in space and time. Thirdly, patients that are treated in a hospital have a well-defined care provider (in Sweden, primary health care organizations).

In a home-based dialysis situation none of those three fundamental assumptions are valid. Thus, from the perspective of an institution (e.g. a hospital) we might encounter several types of changes and challenges, such as: distribution, expansion, and creation

of institutions (see Section 4 for a more detailed description). We are currently focusing on the first and third type of change. For instance, we believe that the patient has to be responsible for parts for their treatment at home. The possibilities of shared responsibilities between institutions and the care receiver are, we believe, fundamental. But we must also include new responsibilities from home care units (in Sweden, secondary health care organizations).

Smart Care Scenario In this project we address another aspect of electronic health care. Again, due to an aging population, there is a need to allow elderly people to continue living in their own homes for as long as possible. The suggested approach is to allow people with special needs independent living by offering the necessary support. Advances in information technology that enable a new way of living come in two varieties.

Firstly, we have the technological means to support smart homes and to connect smart equipment and people in networks. Secondly, we have advancements in nanotechnology providing us with completely new means of monitoring and supporting life-critical processes in the human body. These measurements can then be communicated to the outside world, or be used by local actuators. In short, we have environment- and human-centric information networks that can interact. Evidently, in order to cope with this type of complexity, we have to create a scalable institution, or ecosystem, centred around the care receiver. Again, issues such as ownership and responsibility of services, as well as accessibility of information, are crucial concepts supporting trust in this type of system.

Experiments Health care can be viewed as an institution (e.g. a hospital) that owns a number of entities (health care functions). Furthermore, these entities provide certain services to a number of users (patients). This perspective on the hospital as an institution can be considered to be of a localized nature, i.e. the patients are situated inside the actual hospital and are taken care of by the hospital staff themselves. However, as we have previously described in the home dialysis and smart care scenarios, a need to move the patients from the hospital to their own homes has started to emerge. This emergent change of the hospital as an institution requires that we change our perspective on things in a quite radical manner, at least when it comes to the implicit consequences concerning concepts such as trust, ownership, responsibility, and accessibility.

We envisage a new model of the hospital as being of an adaptive nature, i.e. the institution must be able to cope with the fact that patients would no longer be physically located at the hospital, but rather in their own homes. What we are primarily aiming at is an institution that successfully is able to transfer the notion of trust, that a patient would have in the services provided by the traditional notion of a hospital, into the new model of an adaptive institution. In this model, the hospital is still considered to be an institution that owns a number of services that offer health care functions to the patients physically situated in the hospital. However, the adaptive institution must also be able to provide the "remote" patients with some sort of health care functions.

In line with the discussion above we are mainly interested in understanding what effects the distribution of a previously centralized institution might have on the operationalization of trust. Especially concerning:

- What criteria are imposed on computational ecosystems in societal settings, such as the operationalization of trust in services provided by a hospital (see Section 3 and 4).
- The reallocation of tasks, previously associated with a centralized institution (see Section 6).

We wish to answer the questions above and conduct a more in-depth analysis of their consequences by future experiments based on a software platform called SOLACE that we are currently developing (see Section 5). These experiments also require a system design of the actual implementation - a computational ecosystem focusing on the health care of a patient and the involved institutions and health care functions.

3 Operationalization of Trust

The concept of trust is a crucial concept in computational ecosystems [3]. The concept itself is very context sensitive. In every situation trust is unique and dependable on a complex set of aspects. In our applications we primarily relate trust to the following four issues: commitment to rules, frequency of positive exchange between two entities, propagation, and context experience. In this chapter we will mainly focus on commitment to rules and artefact-mediated services, since this is very much related to the notion of power and purpose of institutions and change.

In order to create trust in a relation between two entities it is imperative that both entities acknowledge their commitment to the rules of the system authorities, i.e. they are aware of the fact that service exchanges of a negative nature are not tolerated by the system authorities. These system authorities correspond to what we would denote as the primitive institutions of the system, i.e. all entities in the system trust these institutions and their roles in upholding the stability of the system. Therefore, commitment to the rules defined by the institutions can possibly be seen as the necessary price to pay by the entities in order to be certain that there are at least some entities in the system that can always be trusted.

We call this basic approach of trust enforcement in a system institutionalized power, i.e. in exchange for some parts of the freedom of an entity it is assured that there is some other entity in the system that it can trust to look after its interests. An example of institutionalized power can be found in institutions of our society, such as the Swedish national bank. Through proper delegation of authority we can trust a civil servant to perform a certain task for us, e.g., we accept bank-notes from a clerk in a bank office as a payment of a check. We can trust an entity's ability and willingness to perform a task or we can trust the validity of the information transmitted.

We model the concept of trust in institutionalized power in our setting as a 4-place relation, see Equation 1. There are several attempts at giving a formal treatment of and providing semantics for such types of trust relations [4]. In the context of institutions and computational ecosystems, a task is typically performed by some entity that is part of a value-chain, and thus, we have Equation 2 as the implicitly operationalized trust relation involving the artefact-mediated services of a value-chain.

$$trust :< person, person, task, context > \tag{1}$$

$$trust :< person, entity, task, value - chain > \quad (2)$$

The operationalized trust enforcement in Equation 2 corresponds to our everyday trust in an automatic teller machine (ATM) in the wall of a bank building, where we insert our ATM card and provide our pin code and accept the paper money eventually delivered from the machine. We trust that the money are valid and that the transactions on our accounts are reliable. We can have some assurance on the later part by asking for a receipt from the ATM.

We believe that the trust relation modelled in Equation 1 has to be earned and can not be completely engineered into a system. Instead we try to enforce trust through partial operationalization. Our scenarios have helped us to identify some operationalizations of trust, that is mechanisms that can support creation and maintenance of trust in artefact-mediated services. One such mechanism is *explanation*, c.f. the receipt of the transaction described above. The corresponding requirement on the information system is *accessibility*. The accessibility mechanism supports management of services as well as presenting different viewpoints on the information. Technically speaking, we must develop suitable ontologies to support these different point of views. Other examples of partial operationalization of trust are mechanisms developed to protect security and authentication, support non-repudiation, and so forth. We argue that accessibility is a cornerstone of the basic information systems derived from the scenarios above. Other requirements concern *ownership* (supporting accountability and liability) and *responsibility*. We have summarized these requirements as a structural model of a computational ecosystem called ORA. By clarifying concepts such as ownership, responsibility, and accessibility (the ORA model) of an entity we can support different models of trust enforcing mechanisms concerning institutionalized purpose and power [5]. The ORA model strives to address all of these issues in terms of ownership conditions, responsibility contracts, and accessibility manifestation and signaling.

We claim that in order to create trust in artefact-mediated services, some of which can be life-dependant, we have to identify partial operationalization of trust that can be supported by the computational ecosystem. Genuine trust can hopefully be earned by careful design of usage rules and trustworthy deployment and maintenance of the involved services. A basic criterion imposed on a computational ecosystem is accessibility, since it assures proper deployment, finding, and combinations of mediated services. These mechanisms pre-suppose context dependent information about the entities and their services which can support, for instance, explanations. The criteria of ownership and responsibility allow us to operationalize these real world concepts in a computational ecosystem. The addition of the last two requirements thus extends the computational ecosystem from a service-oriented information system, providing trusted functionality, to a trusted system in the real world. The next three paragraphs elaborate on these lines of reasoning.

Ownership One important aspect of trust is that it implies a relation between a person and an entity (Equation 2). The entity referred to in the relation is an embodiment of a concept (e.g. a service) that exists in an artificial environment. Furthermore, between a person and an entity there must also exist a condition that proves the validity of the ownership. There are many different ways of enforcing this relation. We suggest that

one way of handling the enforcement of this relation is to focus on who actually owns an entity in the real world. The concept of ownership renders itself as a certain relation:

$$ownership :< person, entity, condition > \quad (3)$$

Responsibility The concept of ownership in the ORA model is obviously of great importance, since without it the notion of trust a user puts in an institution and its constituents will be difficult to enforce. Especially without a mechanism that couples the real world and an artificial environment. However, by incorporating ownership as a key concept in the ORA model it is also very important that we introduce another concept that is very much related to the concept of ownership, namely responsibility. The reason for this is that if an entity is not owned by anybody, its responsibilities towards an accessing party in a societal setting cannot be enforced in a legal fashion. The issue can be perceived from two different perspectives.

The first perspective is that of the owner (individual or organization) of an entity. He or she offers a set of information or functionality to an accessing party using the entity in question. The owner of the entity has the right to require a mutual understanding in the form of a legally binding contract. The contract outlines the responsibilities of the entity and consequently also the responsibilities of the entity owner. If the entity in question does not fulfill its responsibilities in accordance with the contract the negatively affected party has the right to take legal action.

The second perspective is that of the user of an entity. The user is willing to access a set of information or functionality offered by a second entity that is owned by someone (i.e. individual or organization). However, due to the fact that there is a possibility that the actually accessed set of information or functionality does not match the offered set of information or functionality, a contract is required between the user and the owner of the accessed entity. As a consequence of these two perspectives the concept of responsibility can be viewed as a societal relation between two entities in the form of a legally binding agreement:

$$responsibility :< entity, entity, contract > \quad (4)$$

Accessibility The two concepts of ownership and responsibility can be viewed as two cornerstones in the ORA model concerning the support for trust enforcing mechanisms. However, yet another, and more, fundamental concept needs to be considered in order to enable ownership and responsibility, namely accessibility. If an entity is supposed to access another entity there are two important issues involved that have to be properly handled.

Firstly, in order for one entity to interact with another entity, there has to be some way for the first entity to identify the other entity. We argue that all entities can be considered to make use of something we would denote as manifestation in order to handle this. The concept of manifestation is twofold, it refers to the fact that an entity actually exists (both in the real world as well as in the artificial environment), but also to the fact that an entity must in some way be perceivable by other entities inhabiting its environment. Secondly, once an entity has found another entity it must know the nature of its interface in order to be able to communicate with it. Since an entity makes use of

such a general concept as manifestation in order to identify potential entities to interact with, it is not clear, at least not from a computational perspective, what interface a newly found entity makes use of.

The only way to solve this issue is that all entities agree to make use of the same primitive communication interface and medium, no matter what manifestation they have chosen to make use of. We call this primitive communication mechanism signaling. Thus, in order to support the concept of accessibility in a computational ecosystem it is important to realize that manifestation and signaling are fundamental parts of the relation between two entities:

$$accessibility :< entity, entity, manifestation, signaling > \quad (5)$$

The concept of accessibility is mainly based on theories in cognitive systems, e.g. autopoiesis. We believe that if a system and its structure is to be perceived, understood, and possible to relate to by a human user, it is important that we make use of related structures. In autopoietic systems' theory the notion of an operationally closed observer, as well as relational domains and the physical space, is of fundamental importance [6, 7]. Therefore, we try to model our computational ecosystem (i.e. a system that conforms to the ORA model) using similar notions, such as manifestation and signaling, that are supposedly the most primitive mechanisms for observation and communication.

4 Institutions in Change

The introduction of artefact-mediated services in commerce, business, and health care (i.e. the introduction of corresponding electronic institutions) pose a challenge of change in "the ways things are done here" and hence also a challenge when it comes to overall maintenance and creation of trust concerning the new services provided. For example, when ATMs were embedded into the physical walls of bank buildings, this enforced the institutional power of the banks. The physical embodiment of the new service simplified trust and acceptance of it since the new service was an obvious (physical) connection to a trusted institution.

However, our home dialysis and smart home scenarios are fundamentally different. For example, in the home dialysis scenario we have a transition from a centralized institution to a distributed institution, coupled with a redistribution of tasks and responsibilities. Since trust is a very context dependent concept it can not automatically be subdivided or allocated to subparts of the involved services/devices. Instead we have to recreate and enforce the concept in the distributed institution. Using partial operationalization of trust, this comes down to starting with a reassessment of tasks and roles in the original institution prior to any new distribution of tasks and roles, and prior to assignments of tasks that can be electronically supported in the new distributed institution.

In the home dialysis example the original institution is a hospital to which patients come regularly for their dialysis sessions. The new distributed institution consists of parts of the hospital, parts of a local home care provider and parts of the care receiver's home. A major factor behind the introduction of home dialysis is that a patient's quality of life can be greatly improved in most cases by enabling more frequent but shorter dialysis sessions, compared to the normal case where we have a continuous mode of

operation. In our case, a first attempt by the hospital to introduce home dialysis was to install dialysis equipment as well as a quite advanced video conferencing system in a home. This solution turned out to be a failure, mainly due to three reasons.

Firstly, it was difficult for the nurses at the hospital to take responsibility for tasks they did not control locally. Secondly, the patient did not have trust in the treatment despite he/she could have a tele-presence of the nurses. Thirdly, in the end a nurse had to visit the patient regularly, which increased the cost of the treatment. This example also illustrates a difference in tele-medicine tasks, such as surgery on a distance, and home dialysis. In the former case we have a situation similar to the ATM example above. In the latter case we have to change the institution into a computational ecosystem supporting electronic health care with trusted artefact-mediated services.

A closer assessment of the culture and tasks associated with dialysis in hospitals revealed the following: as preparation for a dialysis session a nurse performs a set of different tasks, among them he or she takes a blood pressure test and weighs the patient. Often the nurse also pinches the patient gently. It turns out that these tasks are performed in order to determine the "wet-weight" of the patient, e.g., to decide how much dialysis to perform in order to get the "dry-weight" of the patient. As part of the culture in hospitals there is a rather strict order of roles and responsibilities in the tasks performed. One of these cultural aspects is typically that the patient has the sole role of being treated with no obligations or responsibilities. In our case it turns out that a patient can learn his wet-weight and thus know how much dialysis he or she needs. Given this knowledge the patient will most often be confident enough to personally take responsibility of this task.

In our situation we have thus found a set of tasks that can be difficult to perform over a distance, since the actual body of the patient is involved. Furthermore, as a result from proper training, the patient can in many cases take responsibility for certain tasks, which in turn increases his or her trust in the possibilities of home dialysis. During dialysis, the most critical situations occur when the patient has a sudden drop of blood pressure. In that situation the patient needs the assistance of a nurse in order to manage the dialysis. Clinical tests confirm, however, that this kind of problem is mainly due to stress felt by the patients.

Again, an educated patient in a home situation is very unlikely to experience this kind of problem. As a result of the previous discussion, but also as an attempt of formalization, we model the concept of an institution as a 5-place relation involving people, tasks, roles, culture and the context of an institution (Equation 6).

$$institution :< people, tasks, roles, culture, context > \quad (6)$$

$$context :< institutionpurpose, powerrelations, norms, locations > \quad (7)$$

$$task :< ownership, responsibility, accessibility, purpose > \quad (8)$$

Furthermore, in order to clarify the notions of context and tasks we model those concepts as two 4-place relations (Equation 7 and 8). Please note that *context* in Equation 6 and 7 refers to the context of the institution. In summary, preserving the trust in institutions and associated services, when they are transformed into computational ecosystems, can be quite challenging. For instance, consider the following types of institution changes:

- *Distribution*. The distribution of an institution typically implies an expansion of business at new sites or focusing parts of the core business to different locations, i.e., distribution of location in the context relation above. Trust is enforced at least initially due to the positive meanings of (successful) expansion or (quality assuring) focusing.
- *Expansion*. The expansion of an institution typically means an introduction of new services. The acceptance and trust in the new services mostly depend on how strongly related the new services are to the power relations of the institution.
- *Creation*. When different aspects of institutions aiming at the same market are combined a "hub" or "portal" is created. Ecosystems typically also evolve over time. Institutions and/or services might come or leave as new value-chains are created or changed.

We cannot expect an obvious structural mapping from trust in the services provided by an institution to the case where we distribute the institution and its related services. As we have illustrated in our home dialysis scenario we typically have to carefully reassess tasks and culture in the original institutions before we attempt to redistribute and coordinate the involved tasks. Furthermore, in order to support trust in the computational ecosystem we have to localize responsibilities in order to minimize "responsibility-over-distance".

5 Experiment Infrastructure and Architecture

According to our previous definitions concerning institutions, contexts, and tasks, a computational ecosystem can be defined as the union of institutions restricted to services and service chains, typically involving several institutions. For example the home dialysis ecosystem which has the following initial structure:

$$hospital|_{dialysis} \cup homecare|_{dialysis} \cup transport|_{dialysis} \tag{9}$$

In order to enforce the notion of trust in computational ecosystems, it is not enough to just model the entities and the coordination of them according to the ORA model or a union of restricted institutions.

We must also address the notion of trust in terms of a supporting architecture and infrastructure in order to offer a basic structure and methodology for the modeling and implementation of entities in a computational ecosystem. The infrastructure consists of a number of primitive entities and system functions (corresponding to primitive institutions) that need to exist in order to enforce the purpose and goal of the ORA model and consequently also the implied architecture. In summary, we need both an infrastructure and an architecture in order to handle methodological issues of computational ecosystems.

Infrastructure From a system perspective, a computational ecosystem is constituted by a number of entities that fulfill the concepts outlined by the ORA model, i.e. ownership, responsibility, and accessibility. However, by fulfilling these concepts we have also implicitly stated that there has to exist some sort of primitive entity that connects the real

world with the artificial world, i.e. the physical environment and the computational environment. The connection manifests itself as a person/organization representative. This type of primitive entity represents a physical individual in the computational ecosystem, and hence if an individual has willingly introduced his/her representative into the system, the rules and norms of the ecosystem explicitly apply not only to the representative in the computational ecosystem but also to the individual/organization in the real environment (responsibility propagation). We denote such a primitive entity *owner representative*.

Another type of primitive entity that must exist in a computational ecosystem is a *portal*. As we have previously described, the ORA model defines accessibility as the basic means of signaling and manifestation of entities in a computational ecosystem. The primary responsibility of a portal is to map the notion of accessibility into a direct access to an entity (the portal could just as well partly prohibit such access, offering a sense of high-level security). In other words, if one entity wishes to find another entity, this is done by posing a query to a portal. The result from such a query is a reference to some set of entities that matches the queried manifestation.

In summary, the primitive entities of a computational ecosystem strives to fulfill the complex mapping between two different aspects of the ORA model: world-to-world mapping and manifestation-to-entity mapping.

Architecture In terms of implementation, manifestation must rely on a common architecture since the intent of an architecture is to support the modeling and deployment of all entities, primitive as well as domain specific. However, an architecture does not by itself enforce the complete model in question. This task must be achieved by the entities themselves. Thus, in our case (considering the ORA model) the architecture must support the existence of entities (accessibility) and the primitive entities must support the world-to-world mapping (ownership and responsibility).[1]

6 Experiment Design

As previously described, an ecosystem can be viewed as a combination of a number of restricted institutions, emphasizing a certain task at hand (Equation 9). In this design example, the emphasized task is home dialysis. Previously, the task of home dialysis was supposedly performed by one single institution (i.e. a hospital) and this institution had the sole responsibility for successfully conducting the task. However, for various reasons, we now want to make use of a number of different institutions that can aid in performing the task at hand (see Fig. 1). All of these institutions have one thing in common, they are physically distributed at different locations, one of them is even mobile (i.e. transportation). The question now is: how do we design a distributed computational ecosystem that enforce and partially operationalize trust? Our initial premises in designing the computational ecosystem are:

[1] We are currently developing a software architecture called SOLACE (Service-Oriented Architecture for Communicating Entities) as well as a number of primitive entities, specifically addressing the ORA model. Contact the authors for more information on this experimental platform.

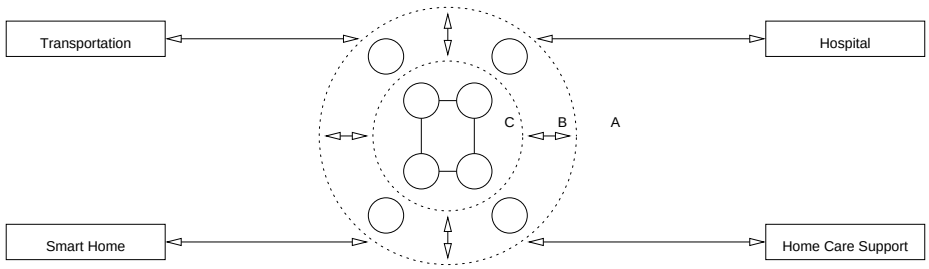


Fig. 2. Conceptual decomposition of services and institutions. The dotted circles correspond to two computational ecosystems B and C that are situated in an open environment A. The high-level services in B are decomposed into sub-tasks in C, but from an implementation perspective all services and corresponding sub-tasks can manifest themselves differently depending on the institution currently offering the service.

- *Value-Chain Consistency*. All tasks involve a number of services, supplied by a certain institution, forming value-chains. These value-chains must never be broken when the structure of an institution changes (e.g. as a result from distribution).
- ORA *Consistency*. All services involved in a value-chain must always be accessible, fulfill some responsibility, and have a clearly stated owner.

In the design process of a computational ecosystem it is important that we first identify the involved services and institutions. It is not until this activity of identification has been successfully performed that a number of tasks will reveal themselves as possible to conduct in a computational manner, hence, forming the basis of a computational ecosystem. The original institution (i.e. the hospital) supplied the care receiver with a certain value-chain (i.e. dialysis).

The main idea at this point is that the original value-chain shall be distributed in such a way that the services can be physically located at the new institutions (e.g. transportation, hospital, smart home, and home care support). Originally, the hospital had the complete responsibility of the value-chain in question, since all services were related to the context of that particular institution. However, if one of the chain's services is relocated into the context of another institution, how will this affect the responsibility of the original institution? We have to further analyze the involved services and see if they are possible to decompose, and then relate the decomposed parts with certain contexts. After this we will be able evaluate their accessibility and consequently also their ownership and responsibility.

We have now decomposed the value-chain into a number of services, that previously were related to the context of the hospital. At this point we move the following services into the context of the smart home (since this is where the care receiver will be situated from now on): machinery setup and care receiver calculations. The reason for this is that they are services related to physical preparations of the session and cannot be performed in a context other than that of the care receiver location, i.e. the smart home.

Obviously it is only the information processing services of the decomposed value-chain that can constitute the computational ecosystem of a dialysis session (see Section 3 Operationalization of Trust for an argumentation concerning operationalization). These services are: care receiver monitoring, machinery monitoring, and dialysis session logging. However, due to the fact that the services fulfill their responsibility over a spatial distance and that information processing in general is involved, a number of opportunities as well as difficulties arise. The opportunities arise since the services and the information they process can be used to produce new information, relevant not only for the hospital staff but also for the care receiver and the smart home.

Hence, new services can be introduced into the computational ecosystem that are of benefit to all parties involved. Furthermore, since the information processing services communicate over a spatial distance the dialysis sessions do not have to be conducted in the hospital, but rather in the context of the smart house. However, the difficulties of changing the role and responsibility of the original institution arise for the same reasons as the opportunities. Monitoring sessions have to ensure connectivity between the involved services as well as institutions involved in the value-chain. Furthermore, security, integrity, and privacy of information must be completely guaranteed.

In section 5 we identified the need for a primitive entity type in computational ecosystems called portal. Such an entity is supposed to handle manifestation-to-entity mapping on request from entities in the system. When it comes to our home health care ecosystem we need four different portals (one per institution): hospital, transportation, smart home, and home care support. Each of these portals will keep track of the entities/services related to that particular institution, and upon a certain request, a given service reference can be supplied to the querying party. However, it is important to notice that a portal does not necessarily only have to handle manifestation-to-entity mapping requests, but could just as well make use of this mechanism to ensure that requirements such as security, integrity, and privacy of information related to the services associated with a particular portal is effectively taken care of.

Associated with each portal, as with all entities in a computational ecosystem, is an owner representative. All of the owner representatives are associated with a person/organization in the real world, and as previously stated, when a owner representative is introduced into the computational ecosystem all rules and norms apply to this entity as well as its associated person/organization in the real world. In the case of the health care ecosystem, we have a minimum of four owner representatives, corresponding to the involved institutions. However, the number of owner representatives could just as well correspond to the number of services involved in the complete ecosystem.

The final type of entities/services that are involved in our health care ecosystem are those that specifically correspond to the information processing services previously described in this section. All of these services can now be associated with their corresponding portal, ensuring proper handling of issues such as security and integrity. Furthermore, by introducing the services into the ecosystem, they must also be associated with a certain owner representative. In effect, if a service present in the computational ecosystem does not fulfill its responsibilities towards a certain person/organization, this can be traced back to that particular entity's owner representative, and consequently also its corresponding person/organization in the real world. We believe that this chain

of responsibilities and ownerships can be used as a fundamental trust enforcing mechanism.

7 Concluding Remarks and Future Work

Recently a new and important domain for applied information technology has emerged, namely electronic health care. The focus of this domain is to cope with an increasing demand of health care services and at the same time manage the increasing costs in the social and health care systems without sacrificing the quality of services. As we have previously described in this chapter, this involves the distribution of a previously centralized institution. However, in doing so it is very important that we do not overlook the fact that users can be negatively affected by the involved changes. In effect, we argue that the users' trust in the involved information systems (i.e. computational ecosystems) is of crucial importance, if the forthcoming changes are to be accepted by the affected users.

Primarily, what we need is an understanding of the operationalization of trust. We have chosen to approach this issue in terms of a model comprising fundamental societal concepts such as ownership, responsibility, and accessibility (i.e. the ORA model). We have used these concepts and their implications to understand in what way the reallocation of tasks and services previously related to a centralized institution will affect trust in the corresponding distributed institution.

We claim that in order to retain trust in the distributed institution we have to identify partial operationalization of trust that can be supported by the computational ecosystem. In summary, the basic criterion imposed on a computational ecosystem is accessibility, since it assures proper deployment, finding, and combinations of mediated services. Furthermore, the criteria of ownership and responsibility allow us to operationalize "real world" concepts in a computational ecosystem. The addition of the last two requirements thus extends the computational ecosystem from a service-oriented information system, providing trusted functionality, to a trusted system in the real world.

The concept of trust is complex and involves different aspects. In future research we are interested in gaining a better understanding of the relations between real world service operators and users by introducing and discussing concepts such as delegation and mediation. In the context of mediated services this may give us fruitful insights in how to operationalize trust in computational ecosystems aiming towards the challenging area of distributed home health care.

From a more general perspective, we have understood that service-oriented systems, such as computational ecosystems in electronic health care, are of a very "open" nature. We consider these complex systems to be characterized as consisting of a set of distributed autonomous software entities, making use of different communication platforms, and maybe most importantly, considering human users as first-class citizens. At this point we have understood that we seek a model for describing an open environment of service-oriented systems, and that the primary observers of such environments are human. Furthermore, we believe that the algorithmic problem-solving focus in software engineering somewhat hinders us in finding such a model. Turning our attention to a different area of research, namely cognitive theory, with a focus on complex and dy-

namical systems could therefore be fruitful. The reason for this is mainly that complex systems and cognitive theory both consider issues related to observation of systemic properties of non-deterministic environments.

8 Acknowledgements

We would like to acknowledge the importance of sharing ideas and experiences with members of the Societies of Computation research group. We are also very grateful for the valuable information and support from Hans Tap and Bo-Krister Vesterlund, Department of Human Work Science at the Blekinge Institute of Technology, concerning their knowledge about home dialysis. Finally, we should mention the project and involved participants of the European Union Information Societies Technology programme ALFEBIITE.

References

1. Estrin, D., Govindan, R., Heidemann, J.: Embedding the internet: introduction. Commun. ACM **43** (2000) 38–41
2. Gustavsson, R.: Agents with power. Commun. ACM **42** (1999) 41–47
3. Castelfranchi, C., Tan, Y.H.: Trust and deception in virtual societies. Kluwer Academic Publishers, Boston, MA (2001)
4. Jones, A.J.I., Firozabadi, B.S.: On the characterisation of a trusting agent - aspects of a formal approach. In: Trust and deception in virtual societies. Kluwer Academic Publishers, Norwell, MA, USA (2001) 157–168
5. Gustavsson, R., Fredriksson, M.: Coordination and control in computational ecosystems: a vision of the future. In Omicini, A., Klusch, M., Zambonelli, F., Tolksdorf, R., eds.: Coordination of Internet Agents: Models, Technologies, and Applications. Springer-Verlag, London, UK (2001) 443–469
6. Maturana, H.R.: Cognition. In: Wahrnehmung und Kommunikation, Peter Lang, Franfurt (1978) 29–49
7. Maturana, H.R.: Reality:the search for objectivity or the quest for a compelling argument. The Irish Journal of Psychology **9** (1988) 25–82

Paper 2: Why Trust is Hard - Challenges in e-Mediated Services

Rindebäck, C & R. Gustavsson (2005)

Appears in:
Social Order in Multiagent Systems. Falcone, R., Barber, S., Sabater-Mir, J., and Singh M. p.180–199.
Springer Berlin / Heidelberg

Why Trust is Hard - Challenges in e-Mediated Services

Christer Rindebäck, Rune Gustavsson

School of Engineering, Blekinge Institute of Technology
S-372 25 Ronneby, Sweden
{christer.rindeback,rune.gustavsson}@bth.se

Abstract. Design and maintenance of trustworthy electronically mediated services is a major challenge in supporting trust of future information systems supporting e-commerce as well as safety critical systems in our society. We propose a framework supporting a principled life cycle of e-services. Our application domain is distributed health care systems. We also include comparisons with other relevant approaches from trust in e-commerce and trust in agents.

1 Background

Trust has been identified to be a key issue when it comes to the design of user-accepted behavior of complex computer systems [1]. Examples of such systems include Multi agent systems (MAS) and emergent systems such as Network Enabled Capabilities (NEC) in defense and efforts related to European EC Programmes in Ambient Intelligent Systems (AmI). Furthermore R&D efforts in GRID computing and web services have a clear focus on issues related to design and maintenance of trustworthy information systems. Although trust and trustworthiness are common denominators in those efforts the approaches are quite different illustrating the complexities of the subject matter as well as the different backgrounds. Reputation and brand naming are examples of trust creating signs in the real world. The purpose of our contribution is to combine different approaches toward aspects of trust and trustworthiness into a framework that allows us to have a principled approach toward engineering of trustworthy behavior of computer mediated services (e-services). For instance, these systems need to be designed in a way that allows the involved entities to exchange information securely and in a trusted way, and that tasks can be delegated to parties that can be trusted to perform the task as expected by the delegating party.

E-services is advocated by large industry consortia as well as by international research communities as a promising future paradigm of the on-line environment providing electronically delivered service based on assembling and coordination of other services. A particular important societal application area is the organization of future health care utilizing information technology. We have had several projects focusing on future home health care based on emergent technologies. In the home health care area we are investigating support systems for health care personnel, home care personnel and patients to establish a trusted support for all parties involved in home health care. The application area is rich and challenging with respect to different trust models. Our suggested framework is based on our current understanding of trust aspects related to assessments of our prototypes and projects in the area.

Trust is largely a subjective issue [2]. Actors may trust, for example, a low security system, among other possible reasons, because they do not know better or because they think that security is irrelevant for the particular system. Trust has also a contextual relation to risk assessment. Obviously this assessment is fundamentally different in nature if, e.g., your life, reputation, or (part of) your economy is at stake. But trust is a concept with many dimensions directed toward different objects between multiple actors, e.g. agents.

This contribution focuses on principal challenges regarding understanding, designing, implementing and monitoring *trustworthy* information systems. Most aspects of trustworthy information systems, agent mediated or not, have been addressing trust (risk assessments) related to economic risks (e-commerce) or reputation (privacy concerns related to e-commerce). We are addressing areas where your life might be at stake, i.e., health care in home environments (e-health) or e-services used in emergency situations in our society.

To set the scene; we regard in our setting trust as a relation between a subject and an object regarding the behavior of the object in a given situation (context). The trust evaluation is a subjective assessment of the object behavior (actual or expected) based on the subject's relevant criteria. In the case that the object is an artifact, the subjective assessment can be supported or refuted by the perceived trustworthiness of the system. Since systems are engineered we are looking for design and maintenance criteria that supports (enforces) trustworthiness in our framework.

In the following Section 2 *Trust and Agents - a Background* we investigate the relationship between a number of identified dimensions and corresponding objects of trust and specifically trust in relation to MAS. Thereafter, in section 3 *Why Trust is Harder than Trustworthiness*, we identify the main issues of the paper as well a research agenda toward that end. The following Section 4 *A Framework Enabling Assessing Trustworthiness*, describes our approach in more details. We illustrate our approach with an example in designing trustworthy systems in a following section 5, *Trust in e-Services in Home Health Care*. We conclude the paper with two sections of comparisons with other approaches, *Models of Trust - Other Approaches*, and, *Trust in Agents - Other Approaches*. The final section, *Conclusions and Further Research*, includes self assessments and pointers to further investigations on the important issue of trust in electronically mediated services.

2 Trust and Agents - a Background

During the last decade two complementary views on agents and trust have emerged with roots either in agent technologies or in models of trust. In short, we have witnessed research agendas on aspects of trust, from a *user* point, in behavior of agent systems on the one side or research agendas focusing on models of trust *between agents* in agent societies on the other side. Sometimes it is not entirely clear what the focus is in papers on agents and trust. In this paper we claim that the first view is a sound one where as the latter view is more troublesome given present state-of-the-art in agent technologies and models of trust. To support our claim we first give a short overview of relevant models of trust followed by an (also short) overview of state-of-the-art of agent technologies.

2.1 Models of Trust

Below we give a short overview of contemporary models of trust along two dimensions; (1) subjects/objects of trust, more precisely between humans, human entities and organizations, social/natural order, and, artifacts, (2) dimensions of trust, that is, ethical/moral behavior, professional competence, and specifically concerning artifacts, functionality and reliability. It should be noted that artifacts in this overview corresponds to physical artifacts or embedded software control systems (e.g., a sledge hammer or a VCR). We will return to agent-based artifacts later. The following table (Table 1) captures the relevant relationships marked with references to relevant work.

Subject / Object	**Ethical/moral behavior**	**Professional competence**	**Action fulfillment**	**Reliability**	**Functionality**
Artifacts	N/A	N/A	N/A	Muir	Muir
Humans Deutsch, Rempel et. al.	Barber, Baier	Barber	Gambetta	N/A	N/A
Communities Giddens	Barber	Barber	Gambetta	N/A	N/A
Trust in Social Natural order & Confidence - Barber, Luhmann					

Table 1. A matrix of trust models and their relation to objects and dimensions of trust

Trust is complex not just in the sense that we may speak about what to trust by whom, with regard to who, or what, but also with respect to the dimension of the behavior of the object the subject have trust in. The table above depicts a number of subjects/objects of trust as well as dimensions of trust. For example; a human might trust that another human has professional competence in a specific context, or trust that a VCR has the intended functionality and reliability. However, we do not even think of ethical behavior from a VCR but this has emerged as a major concern regarding downloaded software (spyware and malware). The subjects/objects in a trust relation are actors' (phenomena) involved. In the model of trust presented here four categories of subjects/objects of trust are presented. We can, for instance, investigate the trust of an individual in the behavior of a society or the other way around. Depending on what the subject-object roles are we can have quite different models and outcomes of assessments of the relevant trust example; an illustrative example is the different views and concerns related to privacy in our societies. The following subjects/objects of trust are part of our model:

– *Trust in social/natural order and confidence* - Our society rests on basic assumptions about what will and will not happen in most situations. For instance we have trust in the natural order, that the heaven won't fall down or that the natural laws will cease to be true. There is also a general trust related to the social order in most of our societies, that is that the governmental representatives will do the best for the citizens and countries they represent and follow laws and norms as well as

follow established practices accordingly. This mutual trust isn't something that actors in general reflect consciously about. The non-reflective trust serves as a basic trust/confidence level for our daily actions where in general there isn't any alternatives to the anticipated risks. The notion *confidence* [3] is sometimes used in situations where actors in reality have no choice. It isn't a viable option to stay in bed all day due to concerns about the social or natural order.

- *Trust in communities* - Humans are often part of a larger community. In the society we have for instance companies, non-profit organizations, governmental institutions and other groups of humans, which often act according to policies, and interests of the community. In many cases the trust may be attributed primarily (or at least in part) in the behavior in a community e.g. a hospital. On the other hand, a hospital may be perceived as more trustworthy than another due to better reputation regarding the perceived treatment and quality of their staff. Depending on the context, trust by a subject may be placed on the object being a community, an individual representing the community, or both.
- *Trust in humans* - In many situations we attribute trust toward other humans, we may trust a particular person about his capabilities or trust his intentions about a particular action. When buying a used car for instance we may trust a car salesman to a certain degree or trust a neighbor being an honest person. Trust between humans has been studied among others by [4, 2, 5].
- *Trust in artifacts* - Trust in human made objects such as cars, computers, VCR:s are in some cases discussed in a manner which implies that these objects can be seen as objects in which trust is placed. For instance 'I trust my car' or 'they trusted the bus to arrive on time'. This means that our expectations regarding the objects with respect to reliability are in some sense confused with or attributed for trust in humans enabling the intended behavior (the design and implementation team of a company, the driver of the bus employed by a public transport company). Since it is unusual or questionable to discuss classical (non-software) artifacts as trustworthy entities with bad will (or good) toward others or as in possess of emotions the use of the notion trust in artifacts is not classically applicable in those settings.

The trust by a subject defines toward what object the trust is attributed and along which dimensions. The following classifications of trust dimensions has been identified in literature on trust models:

- *Functionality* - The functionality of an artifact is an important and natural quality of trust, e.g., the tools is expected to function as it should. An implicit trust condition is that the artifact or tool is not behaving in an unexpected or undesired way by its design [6]. As we have indicated earlier, this situation is quite different when it comes to computer (software) based artifacts, that is, e-services. Firstly, the available functionalities, or affordances, are more complex (flexible). Secondly, and more important from a trust perspective the software can be designed by purpose or by affording vulnerabilities to create dysfunctional behavior that can be very harmful to the user or her system. The explicitly available functions and their appearance and accessibility shape the e-service from the perspective of its users. The user has to trust that these services meet her trust criteria in a trustworthy way

without unwanted results. In our framework we indicate how we can meet these requirements from a designers point of view.

- *Reliability* - The reliability of an artifact is another important criteria of trust in classical artifacts. The tools should be resistant to tear and wear in a reasonable way and the VCR should function flawless for some years. Reliability thus means that an artifact can be expected to function according to the presented functionality and is working when needed. In some contexts reliability can be interpreted as safety, for instance, a safe electric equipment has protection (fuses) against short-circuits that could be harmful. Again, when it comes to software mediated services the trust dimensions of reliability and safety need to be assessed from different aspects.
- *Trust in Action Fulfillment* - In cooperation a specific trust dimension surfaces in most contexts. That is, can a subject trust that an object will indeed fulfill a promise or obligation to do a specified action? In a subcontractor scenario or in a health care situation where a doctor has prescribed a treatment to be carried out on a patient by e.g. a nurse concerns may occur about whether the treatment will be carried out or not. Similar concerns can be identified in e-services regarding whether on line banking task will be carried out or not or if a certain ordered good will be delivered or not. Gambetta [2] defines trust as:
  "a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action".
  An actor or group of actors can be trusted to carry out particular actions based on our expectations on action fulfillment. If the a satisfactionary level of trust isn't reached the task may not be delegated to the actor(s) in question.
- *Trust in Professional Competence* - When a decision to delegate a task to another actor is taken this decision is often based on a perception of that actors professional competence. This refers to expectations about the professional abilities [7] of e.g. a doctor or banker and suggests further refinements of trust expectations. We can trust in somebody to have the right competence for carrying out actions associated with their profession. We trust doctors' judgments about medical needs and we trust them in their ability to adjust treatments in accordance with new findings within their area of expertise. In many situations humans can't gain complete insight in all qualities, aspects and problems characterizing professionalism in certain domains where we need help or assistance. Instead the decision weather or not to engage in a relationship with, i.e., a doctor or act according to the recommendations by a professional is based on trust in the professional competence of that actor.
- *Trust in Ethical/moral Behavior* - Trust isn't only related to professionalism in dealing with tasks as such, it is also suggested to be linked to values and less tangible nuances such as ethical and moral premises. If a trusted professional for instance acts in a manner that is perceived as being against common ethical and moral norms we can choose to distrust this person in a given context despite his professional skills. Examples include certain types of medical experiments or other acts that can be regarded as unethical or even criminal if detected. Trust in moral or ethical behavior is, of course, very context dependent. Moral trust or as it is put forward by Barber as trust in fiduciary obligations [7] means that some others in

our social relationships have moral obligations and responsibility to demonstrate a special concern for other's interests above their own. The lack of control will give the trustee the possibility to exploit or harm the truster [8]. The ethical/moral trust dimension is based on a scenario when there is a risk for betrayal based on ethical and moral reasons. For instance in an e-service the information handled by the involved organization about individual clients can be misused in unethical manners in a way that is generally perceived as unmoral and would harm the truster. This is also connected to willingness from the trustee to put the truster's interest before his or her own. For instance, an e-service designed for health monitoring is expected to be mainly beneficial for its customers. The data collected could be used by the service provider as statistical data that could be passed on to the highest bidding parties.

2.2 Models of Agent Capabilities

We note the obvious fact that most contemporary trust models are related to trust and trust dimensions in human-human relations, see previous section. In the same way, trust models related to artifacts are complementary in the sense that human capabilities and expectations in the form of ethical/moral behavior, professional competence and action fulfillment is replaced by the technical requirements of functionality and reliability. For instance a human can be trusted to act in a moral ethical manner in a certain context whilst it makes no sense to claim that an artifact is acting by itself in this manner [9]. Having said that, there is many open issues related to trust in e-services and software agent-mediated services. For instance, can an (agent-mediated) artifact be instructed or designed in a way that measures up to or comprises ethical/moral behavior? The purpose of this section is to revisit the (classical) discussion on trust as summarized in Table 1 into the situations where we have either a subject assessing her trust in agent mediated services or in situations where it is justified to model trust within the (software) agent societies. This is done in our proposed extension table 2. The Subject/Object heading indicates the two different interpretations of trust. The first row of the table is the situation where a user can assess her trust dimensions regarding the behavior of the agent-mediated services offered. The second row is the second reading of trust; between the software agents themselves. The bottom line is that we regard the first user assessed trust dimensions to be the only viable stance given state-of-the-art agent technologies today and in a foreseeable future. Our framework supporting design and maintenance of trustworthy systems is based on that assumption. The rationale for the

Subject / Object	**Ethical/moral behavior**	**Professional competence**	**Action fulfillment**	**Functionality**	Reliability
Agent system behavior	Yes	Knowledge Based Systems	Yes	Yes	Possible
Within agent systems	NO	Possible	Possible	Possible	Possible

Table 2. A matrix of trust models related to agent mediated services and within agent societies

statements of the matrix of table 2 are as follows [10–13], where [11] includes a state-of-the-art overview of agent technologies and a road map up to 2009 and onwards. The different definitions of an agent from the agent research communities are emphasizing an agent as an autonomous computation software entity with a rational behavior. For multi-agent systems the focus is on interactions and co-ordinations of individual agents to achieve a common task or behavior. The capabilities of individual agents and a MAS are determined by which architectures that can be implemented [13, 14]. The capabilities of individual agents have hitherto been focusing on problem solving capabilities manifested by the well-known BDI (Beliefs-Desires-Intention) architecture.

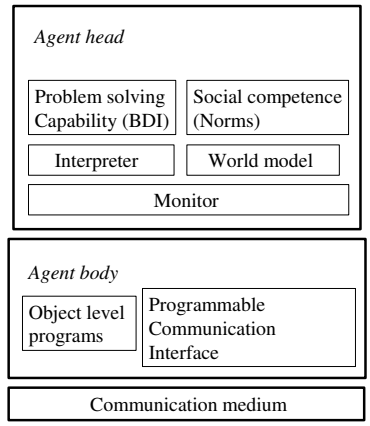


Fig. 1. A reference architecture for agents in a multi-agent system (Belief-Desire-Intention) architecture that also implements a local rational behavior accordingly. Technologies supporting MAS are focusing on Agent Communication Languages (ACL) and coordination patterns as well as community (institution, society) models [11, 13]. The latter are at present based on natural and social systems (normative behaviors).

A minor upgrade of the traditional BDI agent architecture given in [14]. Figure 1 summarizes the current state-of-the-art of agent (head-body) architectures (i.e., capabilities that can be implemented by individual agents or in a MAS). We argue that state-of-the-art agent technologies allow us to have trust in professional competence, and action fulfillment in the behavior of MAS as indicated in Table 2. Examples include knowledge-based systems with explanation capabilities. Furthermore, we can implement a MAS is such a way that the system will indeed have the desired functionality and reliability [11, 12]. However, it should be noted that this goal is not yet achieved concerning reliability but rather a stated goal of the road map of [11]. In that road map, reliability is especially addressing security concerns of MAS. As a matter of fact it limits trust concerns and hence building and maintaining trustworthy systems to issues related to reputation mechanisms, reliability testing, security and verifiability, and electronic contracts. Our framework thus includes and extends issues related to trustworthy

systems as expressed in the road map of [11]. Precisely, for that reason we argue that indeed it is possible to have a grounded belief of trustworthiness by the user in agent based behavior such as e-services. That is, an example of ethical/moral behavior of agent systems in table 2.

On the other hand we argue that trust within agent systems in the sense of trust equal to the phenomena of human trust is beyond state-of-the-art in a foreseeable future mainly due to the fact that we do not have a corresponding complementary component (e.g., consciousness) complementing the architectural components Problem solving capability and Social competence of Figure 1. Regarding the other qualities within an agent society such as professional competence those qualities requires implemented mechanisms supporting self-adjustments, negotiations, learning, and semantic control. Those and similar mechanisms can be available within the next 5-10 years [11]. We will return to some of those topics in Section 7 on Trust in agents - other approaches.

3 Why Trust is Harder than Trustworthiness

We model trust in e-services as an individual assessment of trustworthiness of that service taking into account the given context. Our approach toward enabling trust by users and societies in e-services is consequently to focus on designing and building trustworthy systems based on a principled approach of handling trust concerns of system actors, e.g. users, and transforming those concerns into design principles and signs to be assessed by the users evaluating the trustworthiness. Our framework identifies a conceptual structure and some important processes toward a methodology to that end.

E-services is not a unambiguously defined concept [15] but a common definition is: ”Interactive software-based information systems received via the Internet” [16]. The information system typically involves many system components, e.g. software and artifacts. Typically e-services are composed of other services provided by third parties. For instance in order to distribute sensible health care data a suitable certificate may be used to create the necessary trust in the service.

When buying anti-virus software we are rather buying a service than a product. The software is bought with an initial subscription. When new viruses are discovered information about the viruses is added to a database that supports downloading upgrades to subscribers of the service. This service oriented approach also leads to a continuous relationship between the service provider and it’s users. In health care we are seeing similar tendencies where patients are treated over longer time spans compared to earlier than they just visited hospitals when they were ill and left upon recovery. Recovery and care will to a larger extent take place in the home of those needing care assisted by health care personnel.

The structure of the framework, i.e., the basic concepts and their relations are described in Section 4. In our model we take, as have earlier been said, into account relevant trust concerns, aspects, mechanisms and signs supporting user’s trust assessment. Further details on that strand are given in the next section. Needless to say, much research and experiments remains to be done to assess and refine our approach to meet the goals expressed above.

The inherent difficulty with qualities such as trust, and other related qualities such as security, privacy, and usefulness, is its *systemic* nature. That is, these qualities can only be assessed at the system level. From an engineering point of view these systemic qualities are sometimes called *non-functional* because:

- The quality cannot be decomposed into qualities of components.
- Two components can have the quality but not their composition. This aspect is particularly important in the area of composition of e-services. Reasons behind this non-compositional nature of systemic qualities include loss of quality due to uncontrolled (unforeseen) interactions between components or due to incomprehensible complexity of the conjunction of services perceived by the user.

An example of a systemic quality is traffic security. We have learned that by engineering vehicles taking into account traffic security *concerns* (expressing aspects that are manifested in *mechanisms* such as reliable brakes, air bags, belts, and crash zones) the risk assessment by users are simplified to assessing *signs* (mediated through brand names, accident statistics, or reputation) *associated* to the vehicle. The society has on its side developed an infrastructure supported by another set of traffic security concerns (road systems minimizing collisions, vehicle control authorities, education, monitoring authorities, legal frameworks) aiming at a higher traffic security in the society at hand. We all know that accidents still cannot be avoided but still we all trust the traffic system enough to use it on a daily base at our own decisions. Of course, the efforts of creating a trustworthy traffic system are ongoing processes with the explicit and *measurable systemic goal* to decrease the numbers and the severities of accidents. In effect, our societies have identified, since the last century, a set of traffic security concerns and aspects that have been translated into mechanisms implemented in different subsystems (components), i.e., more trustworthy vehicles, safer roads, and better monitoring measures. Our societies thus have furthermore developed a strategy and means toward attaining trustworthy traffic systems that each user can decide to trust (or not) at their will to use in an appropriate way. Of course, nobody believes that building and testing trustworthy components in itself will replace continuous traffic security assessments at the system level. The aim of our contribution is to propose and illustrate a similar comprehensive approach, as in the traffic example, toward supporting trust in e-services. In short, from an engineering perspective, we can only aim at designing, implementing, and maintaining trustworthy systems and components. Our success in gaining acceptance and trust by users of the systems will depend upon how well we have succeeded in translating trust concerns into aspects and mechanisms that can be implemented in a trustworthy manner by providing appropriate signs. At this point in time we, however, do not have an appropriate metric on the systemic level (compared to statistics and assessments of traffic accidents) to enable us to claim that we have a good strategy for supporting the users to gain more confidence in their trust assessment of electronically supported services by, e.g., providing more appropriate signs. Thus, making trustworthy information systems is hard but supporting users trust in them is at the moment very much harder. Our contribution is to outline a framework and processes to enable the first concern and to narrowing the latter divide.

Trust and trustworthiness are two notions we need to use wisely in order to emphasize the differences between the two. Trustworthiness is what designers of systems

can implement [17] as mechanisms into the system manifested by an appropriate set of signs. An actor performing risk assessment related to trust then assesses if the system is trustworthy by inspecting the signs. The judgment whether the system is trusted or not is thus taken by the observer or user of the e-service. It is not possible to directly code trust into the system, see our discussion in Section 2. We use the term *actor* to denote stakeholders in the system. This is because of the fact that different actors may have different considerations related to trust affecting the design considerations needing attention [18]. Trust is obviously very context dependent [7]. We can, for instance, have trust in one actor providing an e-service and then not trust the very same actor in another service. We may also lose trust in some services if something disruptive has happened such as introduction of new technologies or unexpected breakdowns. Identification of and maintaining trust aspects that should be *sustainable* during change are in many applications crucial. There are several approaches aiming at modeling non-functional requirements such as trust by introduction of some measurable quality. A problem with this approach is that the qualities identified and measured have turned out to be quite arbitrary [19]. For instance, user satisfaction, being a systemic quality, has been approximated by a set of measurable qualities by different models. To infer user satisfaction relying naively on numeric calculations of numbers related to those measurable qualities could be misleading at least or totally wrong at worst. Furthermore, it has turned out to be difficult to compare different numerical models or to make predictions due to changes in the system. In the context of trust research there is no consensus about what to quantify, measure or investigate in order to reach a conclusion on whether a system is to be considered as trustworthy or not. This state of affairs imposes challenges on system designers in design, development, and use of tools enabling evaluation of computer systems trustworthiness. Our proposed framework and associated processes are steps in that direction. We compare our approach with contemporary approaches toward trust in electronically mediated services or agents in Sections 6 and 7.

4 A Framework Enabling Assessing Trustworthiness

The following Figure 2 captures the main ingredients of our framework supporting design, implementation and monitoring of trustworthy e-services. The components of the framework are:

- The *context*, including: actors, e-services, artifacts, location, and, time. The context also includes other components and factors such as contracts, ownerships, responsibilities, legal frameworks, work practices, and, organizational aspects.
- *Trust concerns* addressed: e.g., loss of life, threat of privacy, loss of money, loss of reputation, responsibilities, or time and duration of engagement.
- *Trust aspects* that can be derived from trust concerns: legal aspects, responsibilities at breakdowns, information integrity, security, privacy aspects, or explanations of functionality.
- *Mechanisms* that implement trust aspects, e.g., explanations, in a trustworthy way.
- *Signs* ensuring correct implementation of trust mechanisms that can be inspected by the observer of the e-service [20].

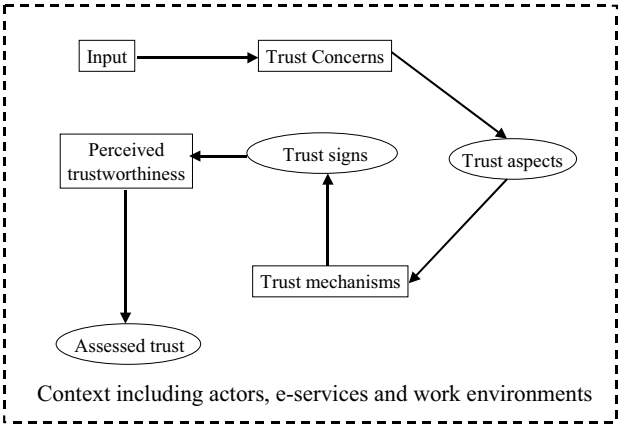


Fig. 2. Main components of our framework supporting design and maintenance of trustworthy electronically mediated services

The relationships depicted by arrows in the figure are typically many - too - many. That is, a trust concern can be broken down into one or many aspects or vice versa. The same argument holds between aspects and mechanisms as well as between mechanisms and signs. An example from the earlier mentioned traffic domain is traffic signs (a mechanism and a sign) that implement trusted traffic information by alternating between sending out red, yellow, or green light. The trustworthiness and trust rely on that all agents involved trust that (almost) all other agents knows the intended reading of the signals and follow suit. As another sign all legal drivers can, on request, provide a valid driver's license.

Another example is security that can be and often is a trust concern. Typically aspects related to security are information security, network security, computer security, or physical security. A typical mechanism related to the aspects of confidentiality, access control and information integrity is encryption. As yet, however, there is no well-accepted sign that encryption has been trustworthy implemented. Certificates - issued by 'trusted third parties' are in many senses to weak today to serve as a trust sign in our setting.

The degree of trust in services possessed by individuals is by no means static. As reported by [5] trust between individuals tends to grow stronger in close relationships, the familiarity factor. Familiarity with services is a strong support of trust. Trust does not only increase it can decline and hence has a dynamic nature [21]. The following factors exemplifies what contributes to the dynamics of trust:

- The actors' experiences through interactions with the e-service and involved actors, experience-based trust [22].
- Changes in society [23]. The climate and attitude toward providers or components of an e-service may change in society in general. If Internet Banks would

be claimed to take to high fees in general the trust in Internet Banks in general may decrease.
- Changes in the composition of services, objects and artifacts. If the composition of an e-service changes, i.e. new technology or a new actor is introduced the trust concerns raised by actors may change.

We have a plentiful of potential changes among the concerns and aspects of trust. In a dynamic society these reasons for changes will prevail. The design of trustworthy e-services therefore is an effort that needs attention not just during the design phase, but also during the whole life cycle of the e-service. The dynamic nature of trust suggest that we continually must re-evaluate and eventually redesign mechanisms and signs of our framework to support efficient and reliable risk assessment concerning trust. Our framework supports this process and is part of our *trust management* process. The following semi-formal notations and definitions clarify the different dependencies of Figure 2 and provide a backbone for our methodology of trust management of e-services. The intended reading is that users express trust concerns in a given context. This input can be translated into a set of trust aspects. One category of trust concerns often mentioned individuals are related to misuse of owned or generated information that might lead to loss of life, loss of freedom, loss of money, loss of reputation, or receiving unwanted commercial offers. The generic term of those concerns is privacy. However, the given context will qualify the aspects (types) of privacy that are relevant for the expressed trust concerns indicated above. The identified aspects can then be translated into relevant mechanisms, e.g., secure end-to-end information exchange between mutually identified and trusted end users, and validated by appropriated signs. The components of an e-service include providers - the provider of the service to the user, third party actors - enabling the creation and distribution of the service, content of the service - the information and products distributed, computer-based artifacts used to provide the service to the user, access points to the service, and implemented trust mechanisms that are coupled to the relevant trust aspects formally defined as:

- e-service = < Provider, Third_Party, Content, Computer_based_artifacts, Access_points, Trust_mechanisms >

Definitions of the concepts Situation, Trustworthiness, and, Trust:

- Situation = < Time-interval, Location >
- Trustworthiness = < e-service, Situation, Context >
- Trust = < User, Trustworthiness, Signs >

A situation is a binary relation between a time-interval and a location (where the service is delivered and used). Trustworthiness of an e-service connects the service to a situation and a context. The context is specified in the design phase of a particular e-service, c.f., our case scenario of Figure 3. Finally, the perceived trust by the user is a three-valued relation connecting the user, and signs that manifests the trustworthiness of the e-service. The value of trust can be of any type that supports reasoning and modeling in the framework. Examples include Boolean values (Yes, No), numerical values modeling strength of Belief in the trust, c.f., [24], or measuring fuzziness. In more elaborated

modeling where partial ordering might be useful we can use lattices as the value domain of Trust. Given those definitions in a formal language we can define and reason about properties and invariance of properties of and between components of our framework in Figure 2 by introducing a suitable logical framework and notations. Given that logical framework we can for instance state precisely what we mean by "Trustworthiness of an e-service independent of a set of situations", "Trustworthiness of an e-service independent of a set of mechanisms" or other invariants by introducing restrictions of formulas over sets.

We will return to the methodological processes related to the structure of framework in Figure 2 later. That is support for design and trust management. Design and trust management is modeled after Boehm's risk driven spiral model[25]. Eventually we hope to supplement the framework with guidelines on how to design and implement mechanisms and signs supporting trustworthiness.

5 Trust in e-Services in Home Health Care

Distributed health care (e-health) utilizing Information and Communication Technology (ICT) is a vibrant area of research and development worldwide. First and foremost there is an international societal-economical need to assess current models of health care. Not the least in health care for people with special needs.

The underlying idea behind e-health in homes is that given the proper support a patient (e.g., elderly person) can stay longer in his/her home and thus have a higher quality of life than otherwise. At the same time the society gains is expected to be lower total costs and fewer burdens on hospitals and other health institutions. E-health systems are typically very complex socio-techno systems and a shift toward future e-health systems requires and understanding of the socio-economical aspects as well as of systemic possibilities and considerations. Systemic *invariants* such as, e.g., "good care" and trust, have to be sustained during introduction of ICT in order for e-health to get acceptance by involved parties We have developed our framework to support design, implementation and maintenance of this change of institutional centric health care into a distributed patient-centric health care while preserving trust in the necessary services by all agents involved, not least by the patients.

The following figure, Figure 3, captures our 'patient-centric view' of e-health. We have investigated this scenario in several national and international projects [1] in distributed health care. A result of those investigations, based on lessons learned and insights, is the framework presented in this paper.

The scenario above involves five *institutions*, three concerning health care and home support services and two service providers. Furthermore we have three types of teams, hospital *teams*, local health care teams, and mobile home support teams. Two portals - e-service systems, including stationary and mobile access points) support the activities of the scenario. The Health care portal provides sensors supporting monitoring of the health of the patient. The related information is transformed into suitable formats for assessments of the teams and the patient in a role-based manner. The smart home portal has sensors and actuators supporting the patient in his daily life at home.

[1] EC Alfebiite - http://alfebiite.ee.ic.aac.uk

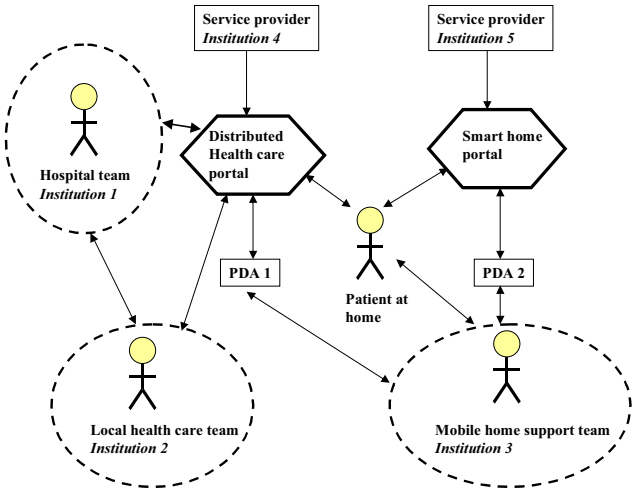


Fig. 3. Teams and institutions involved in distributed home health care

A successful transition from today's health care organizations and practices to the situation depicted in Figure 3 would typically mean that large parts of the treatments of the patient now conducted in hospitals or similar institutions will take place in the homes of the patients. This also implies that the work situation for Institution 3 (Home support team) will be more qualified although their basic education will mainly be as now. The home support team thus needs good support from the artifacts delivered by Institutions 4 and 5 and a mutually trusted case-based delegation of tasks from the hospitals in order to do their job in a satisfactory way. To summarize the new situation: The health care authorities that are responsible for the health care have to have a grounded trust in that the new organization will deliver high-quality health care services in a cost-efficient way. The involved organizations must also have similar beliefs. The persons involved, not the least the patient, must trust that the new work situation will provide sufficient support for the new work flows. Last but not least the overall systemic goal of "good care" has to be maintained during the ICT enabled transformation. We have investigated several different partial scenarios related to the scenario given above. One set of investigations was related to equipment for measuring the health status of patients (related to the Health Care portal of Figure 3). Another set of investigations is related to improving the learning and knowledge sharing in teams utilizing Peer-to-Peer technologies [2] (Institution 2 in Figure 3). A third set of investigations was focusing on issues of shared awareness and work flow management where we have actors from more than one institution, institutions 1 and 2 of the scenario. Two applications [3] in this

[2] E.g. WoundDoc - an information sharing tool for health care personnel
[3] For more information visit http://www.soclab.bth.se

setting are SHINE - Sustaining health and interaction in networked environments - and DICE - Delegation and interaction in care environments. In the DICE application we had doctors and nurses from either of Institutions 1 or 2. Furthermore, we have nursing assistants belonging to either of institutions 1, 2, or 3.

The workflows will typically be supported by digital information management systems with different types of access possibilities. That is, the primary asset is information and a primary concern is trustworthy management of the information. One important aspect of trustworthiness is thus related to dependability (e.g., security, integrity of persons and data, and accessibility). Field personnel using new digital artifacts (DICE) have frequently raised the following trust concerns during our evaluation tests:

- How do we know that we do the right kind of tasks or actions in the right way?
- What happens if something goes wrong?
- Can our employer spy on us or misuse the information the system provides about our work?
- We have a very dynamic environment. Can we have a flexible system taking care of our mentioned concerns?

The system requirements can be formulated as: Trusted role and context based access control to services in e-health. Intuitively we can presuppose that enforcing normative behavior could be a way to support trustworthiness but here we have to strike a balance between being too restrictive and in that way hampering a needed flexibility in the work flow processes (e.g., as in the DICE system). Our approach to achieve flexibility is to identify and implement context dependent normative behavior.

Background material on different aspects of trust was developed in the EC project Alfebiite. In effect: three supporting frameworks of trust: A logical Framework for Norm-Governed Behavior, A Conceptual Framework on Operational Model of Normative Behavior, and, Communicative Acts and Interactions Patterns developed in the project have largely influenced our approach. The following different concepts of trust have been proposed in those deliverables:

- A mere mental attitude (prediction and evaluation toward an other agent;
- A decision to rely upon the other, i.e., an intention to delegate and trust, which makes the truster 'vulnerable';
- A behavior, i.e., the intentional act of trusting, and the consequent relation between the truster and the trustee.

In our case we focus on the latter two concepts, since they open up for a methodological approach toward creating trust. In the conceptual framework we have models connecting trust and delegation (weak or strong). The models presuppose human agents but some models could also be used in the situation of trust in artifacts (which is of our main concern in our investigations). For instance, we model the trust in artifacts as strong delegation. In the same deliverable we also find the notions of internal and external trust useful for our investigations. The concept of a three party relationship based trust model is also very appropriate in our approach.

Another interesting concept for us is Adjustable Social Autonomy [26] modeling time dependent levels of delegation. Especially, we share the beliefs that "A very

good solution (of adjustable social autonomy) is maintaining a high degree of interactivity during the collaboration, providing both the man/delegator/client and the machine/delegee/contractor the possibility of having initiative in interaction and help (mixed initiative) and of adjusting the kind/level of delegation and help, and the degree of autonomy run time. This means that channels and protocols - on the delegator's side - for monitoring (reporting, observing, and inspecting), and for discretion and practical innovation: for both client and contractor channels and protocols are needed for communication and re-negotiation during the role-playing and the task execution". As a matter of fact, our implementation of the DICE system is designed to meet such requirements concerning run-time observations and adjustments of systems.

6 Models of Trust - Other Approaches

One driver behind the interest in trust and e-services is that higher trust in e-service providers are likely to affect the willingness to engage in relationships and utilize the provided services. As a fact, trust has been defined as a willingness to depend or rely on other actors [27]. From the truster's perspective trust is a mechanism used to reduce complexity [3, 23] under situations of risk where we can choose our path of action based on expectations. One model proposed to deal with trust in risky environment such as e-commerce is the model of trust in electronic commerce (MoTech). It aims to explain the factors that affect a person's judgment of an e-commerce site's trustworthiness [22]. MoTech contains of a number of dimensions intended to reflect the stages visitors goes through when exploring an e-commerce website. The dimensions pre-interactional filter, interface properties, informational content and relationship management will be described below. Each of these components addresses factors that have been observed to affect consumers' judgment of an on-line vendor's trustworthiness.

Pre-interactional filters refer to factors that can affect people's perceptions before an e-commerce system has been accessed for the first time. The factors presented are related to user psychology or pre-purchase knowledge. The first group refers to factors such as propensity to trust and trust toward IT in general and the Internet. Pre-purchase knowledge is related to Reputation of the industry, company and Transference (off-line and on-line). The second dimension of MoTech is concerned with interface properties that affect the perception of a website. Here the components are branding and usability. Factors in the branding component are appeal and professionalism. The usability component factors are organization of content, navigation, relevance and reliability. The next dimension, informational content contains components related to competence of the company and the products and services offered and issues regarding security and privacy. The fourth and last dimension reflects the facilitating effect of relevant and personalized vendor-buyer relationship. The components Pre-purchase Interactions and Post-purchase interactions are related to factors such as responsiveness, quality of help and fulfillment.

The model structures e-commerce designers work and give directions toward important trust considerations during the discussed dimensions. In the light of our framework we would interpret the four dimensions or stages as four situations. For instance the pre-interactional stage is the situation before any interaction has taken place with the

e-commerce web site. The factors are related to concerns, aspects, and mechanisms in our framework. The MoTech components privacy and security are trust aspects and the factors proposed are mechanisms such as policy, encryption and contractual terms in our framework. To summarize: we can model the MoTech approach in our framework whereby we also get a more principled approach for evaluation and maintenance. MoTech is developed for e-commerce applications but has also been tested in other contexts such as on-line gambling.

7 Trust in Agents - Other Approaches

Current state of the art tries to capture and reason about norms in agent societies. These so called normative agents trends are the one lying closest to human behavior as of today. Instead of acting based on reactive stimuli or a message related to problem solving a norm based agent can act based on social norms in order to achieve some kind of goal in isolation or in a team. However the state of the art within the area of MAS-architectures and agent models today merely reaches a desired level of a mixture of normative behavior and reflective behavior in key applications, Section 2 and Section 5. Another approach is to view human and computational agents differently. This is especially obvious when relying on some of the more common definitions of the term agent e.g. Woolridge defines and agent to be *"a computer system that is situated in some environment that is capable of autonomous action in its environment in order to meet its design objectives"* [28]. This definition excludes human beings from the agent metaphor at least in a computational setting. This explicit differentiation between human and computational agents opens for our approach of trust in agents, i.e., trust in human-agent interaction. Where the human is the subject and the agent(system) is the object, see Section 2. Models of trust in agent behavior have been an active research area for more than a decade. Different aspects of trust models have been proposed and sometimes implemented. Concerns of trust in agent behavior goes back to mid seventies where the corresponding systems under investigation was expert systems or in a later terminology knowledge based systems. The tasks performed by the systems were knowledge intensive problem solving in areas such as diagnosis, planning, scheduling, and monitoring. The problem solving capability was captured and engineered to mimic human expertise in selected areas. By necessity the knowledge systems had to handle inherent weaknesses such as brittleness and sometimes assessment conflicts between experts. In short, there were concerns by the users how to trust decisions suggested by the systems. The following trust aspect was identified to remedy these concerns. The users requested explanations of the support for the conclusions drawn by the system. Two explanation facilities, or mechanisms, were identified, i.e., answers to the questions "Why?" and "What if?" c.f., our framework in Figure 2. Different strategies of reasoning and implementations of those mechanisms have been evaluated since that time. A good exposition of trust concerns related to explanations and corresponding mechanisms are described in [29]. Below we assess contemporary efforts in designing and building trustworthy agent systems and e-services. In our discussion we frequently refer to concepts from our framework, Figure 2. Furthermore we base our assessments on our discussion in Section 2, Figure 1, and, Table 2.

MAS (Multi Agent Systems) designers and programmers investigate trust due to it's importance in human interpersonal relationships where trust seems to affect how we make decisions about what to delegate and to whom or whey we choose to act in a way rather than another. For instance why do we trust A to do a task for us instead of B? Here reputation has been identified as a major factor to be aware of. Thus by implementing mechanisms into MAS the intention is to create agent-to-agent trust in trade and interaction between agents in the systems ultimately enabling them to act independently of the agent owner and make deals and commit to tasks on behalf of its owner. These kind of trust supports are mechanism-oriented and it is often hard to assess in what ways those mechanisms are related to trust concerns as expressed in our framework, Figure 2.

The Foundation for Intelligent Physical Agents (FIPA) has proposed MAS Security Models. A good overview of relevant material "Specifying Standard Security Mechanisms in Multi-agent Systems" is provided in [30]. From the point of our framework the focus is on mechanisms. The FIPA requirements are collected from a set of scenarios, related to e-commerce, from which a set of security issues is derived. The corresponding architectural elements, or trust aspect in our terminology, are found to be authentication, authorization, integrity, and privacy. Some generic safeguards are the proposed. There is no attempt by FIPA to address the concerns that might lead to the mentioned set of trust aspects. Neither mechanisms nor signs are explicitly addressed. In practice, it might be difficult to assess how well the FIPA efforts supports trust in agent-mediated services in the selected domain e-commerce. Security aspects of trust concerns of agent behavior are addressed by several researchers beside the FIPA efforts. The application area is typically e-commerce [31]. Again most efforts is devoted to discuss similar trust aspects as in the FIPA case but sometimes introducing other mechanisms and sometimes signs (certificates).

A research agenda addressing challenges for trust, fraud, and deception research in multi-agent systems has recently been proposed [32]. The areas identified are: Trust model discrimination, Building reputation without interaction, benchmarking trust modeling algorithms. Trust as measuring a reputation based quantity is also the basic mechanism in studies of objective-trust based agents [33]. The underlying assumptions with these approaches are that, in our terminology, the chain of trust concerns - trust aspects - mechanisms can be compiled into a metric (sign) calculated by an algorithm. These approaches are of cause possible to model in our framework and of relevance in specific circumstances. On the other hand, addressing trust in life threatening situations such as distributed e-health require a more elaborate approach.

Recent advancements in semantic web technologies as well as in web services and semantic Grid computing make introduces the concept of "smart" or "intelligent" services. In our view those kinds of services can and perhaps should be modeled as agent mediated services. This approach allows a fruitful interaction between the high-level agent approach and the bottom-up approach provided by the web service and Grid computing communities. Both communities have their preferred approach toward trustworthy systems with specific advantages and disadvantages. Our framework is aiming at a common ground for designing and maintaining trustworthy intelligent e-services.

8 Conclusions and Further Research

Creating and maintaining information systems that users can decide to trust is a hard challenge. In effect we ask the user to trust, economically and in some cases even with their life, the behavior of electronically mediated services, e-services. To that end we propose a framework and a methodological approach aiming at designing, developing and maintaining trustworthy systems. The framework is based on the idea that trust is a subjective assessment that is highly context dependent. To capture the anatomy of those assessments we introduce the following concepts in our methodology; trust concerns, trust aspects, trust mechanisms, and, signs. Typically, users articulate trust concerns and they look for signs that will assist them in their assessments. Trust aspects are design tools allowing designers to decide proper mechanisms to be implemented and to provide signs that verify that those mechanisms have been properly implemented. Trust concerns thus give insight into hypothetical or validated concerns related to trust among providers, third party actors and users of e-services, hence trust aspects are operationalizations of the different concerns. Trust mechanisms are implemented trust aspects, e.g., explanations, encryption algorithms etc. The signs, trade marks, documentation, certificates, and so on, provide actors and end users with credentials belonging to an e-service enabling the actors to form their judgment of whether or not trust the service.

We have also included a case study to illustrate and validate our framework related to trustworthy e-services. Our applications are primary distributed e-services supporting a patient and associated health care and home care teams from a home-centric point of view. We have chosen this application area for two reasons. Firstly, the application area is of high societal importance worldwide; secondly, the application amply illustrates the different aspects and challenges of trust in artifact-mediated services. In fact your life might depend on some of those services. We have developed our models in different application projects and based our approach on contemporary R&D on trust and trustworthiness.

In our definition of e-services we have taken into account different aspects of their context, i.e., other actors than the user, other e-services, artifacts and contextual qualities such as contracts, ownerships, responsibilities, legal frameworks, work practices, organizational aspects, and, time. Furthermore, we make some comparison of our approach with contemporary approaches toward trust in system behavior from the e-commerce area and the Multi Agent System domain. The approach and models are to a high degree work in progress and will be refined in other upcoming projects where we have to trust artifact-mediated services where life might be at stake.

References

1. Gefen, D., Straub, D.W.: Managing user trust in b2c e-services. e-Service Journal **2** (2003) 7–24
2. Gambetta, D.: Can we trust trust? In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. B. Blackwell, New York (1988)
3. Luhmann, N.: Familiarity, confidence, trust: Problems and alternatives. In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. Basil Blackwell, New York, NY (1988) 94–110

4. Deutsch, M.: The resolution of conflict; constructive and destructive processes. Yale University Press, New Haven,NY. (1973)
5. Rempel, J., Holmes, J., Zanna, M.: Trust in close relationships. Journal of Personality and Social Psychology **49** (1985) 95–112
6. Muir, B.M.: Trust in automation .1. theoretical issues in the study of trust and human intervention in automated systems. Ergonomics **37** (1994) 1905–1922
7. Barber, B.: The logic and limits of trust. Rutgers University Press, New Brunswick, N.J. (1983)
8. Baier, A.: Trust and antitrust. Ethics **96** (1986) 231–260
9. Friedman, B., Kahn, P.H., Howe, D.C.: Trust online. Communications of the ACM **43** (2000) 34–40
10. Luck, M.: Challenges for agent-based computing. Special Issue of Autonomous Agents and Multi-agent Systems **9** (2004) 203–252
11. Luck, M., McBurney, P., Priest, C.: A manifesto for agent technology: Towards next generation computing. Special Issue of Autonomous Agents and Multi-agent Systems **9** (2004) 253–283
12. Zambonelli, F., Omicini, A.: Challenges and research directions in agent-oriented software engineering. Special Issue of Autonomous Agents and Multi-agent Systems **9** (2004)
13. Sierra, C.: Agent-mediated electronic commerce. Special Issue of Autonomous Agents and Multi-agent Systems **9** (2004) 285–301
14. Hægg, S., Ygge, F.: Agent-oriented programming in power distribution automation : an architecture, a language, and their applicability. Dept. of Computer Science Lund University, Lund (1995)
15. Stafford, T.F.: E-services. Communications of the ACM **46** (2003) 27–28
16. Featherman, M.S., Pavlou, P.A.: Predicting e-services adoption: a perceived risk facets perspective. International Journal of Human-Computer Studies **59** (2003) 451–474
17. Sisson, D.: e-commerce: Trust & trustworthiness (2000)
18. Shankar, V., Urban, G.L., Sultan, F.: Online trust: a stakeholder perspective, concepts, implications, and future directions. Journal of Strategic Information Systems **11** (2002) 325–344
19. Kotonya, G., Sommerville, I.: Requirements engineering : processes and techniques. Worldwide series in computer science. J. Wiley, Chichester ; New York (1998)
20. Bacharach, M., Gambetta, D.: Trust as type detection. In Castelfranchi, C., Tan, Y.H., eds.: Trust and deception in virtual societies. Kluwer Academic Publishers, North Holland (2001)
21. mcKnight, H.D., Cummings, L.L., Chervany, N.L.: Initial trust formation in new organizational relationships. Academy of Management Review **23** (1998) 473–490
22. Egger, F.N.: From Interactions to Transactions: Designing the Trust Experience for Business-to-Consumer Electronic Commerce. PhD thesis, Technische Universiteit Eindhoven (2003)
23. Giddens, A.: The consequences of modernity. Polity Press ;, Cambridge (1990)
24. Singh, M.P.: Trustworthy service composition: Challenges and research questions. Trust, Reputation, and Security: Theories and Practice **2631** (2003) 39–52
25. Boehm, B.W.: Software risk management. IEEE Computer Society Press, Washington, D.C. (1989)
26. Falcone, R., Castelfranchi, C.: The human in the loop of a delegated agent: The theory of adjustable social autonomy. Ieee Transactions on Systems Man and Cybernetics Part a-Systems and Humans **31** (2001) 406–418
27. Dobing, B.R.: Building trust in user-analyst relationships. Ph. d., University of Minnesota (1993)
28. Wooldridge, M.: Introduction to MultiAgent Systems. Wiley (2002)
29. Shapiro, S.C.: Encyclopedia of artificial intelligence. 2nd edn. Wiley, New York (1992)
30. Poslad, S., Charlton, P., Calisti, M.: Specifying standard security mechanisms in multi-agent systems. Trust, Reputation, and Security: Theories and Practice **2631** (2003) 163–176

31. Tan, J.J., Titkov, L., Poslad, S.: Securing agent-based e-banking services. Trust, Reputation, and Security: Theories and Practice **2631** (2003) 148–162
32. Barber, K.S., Fullam, K., Kim, J.: Challenges for trust, fraud and deception research in multi-agent systems. Trust, Reputation, and Security: Theories and Practice **2631** (2003) 8–14
33. Witkowski, M., Aritikis, A., Pitt, J.: Trust and cooperation in a trading society of objective-trust based agents. In Falcone, R., Singh, M., Tan, Y.H., eds.: Worskhop on Deception, Fraud, and Trust in Agent Societies, Barcelona, National Reserach Council, Institute of Psychology, Rome Italy (2000) 127 – 136

Paper 3: Design and Maintenance of Trustworthy e-Services: Introducing a trust management cycle

Rindebäck, C. & R. Gustavsson
In the Proceedings of the Second International Conference on Web Information Systems and Technologies, Setubal, Portugal.
Insticc Press.

Design and Maintenance of Trustworthy e-Services: Introducing a trust management cycle

Christer Rindebäck and Rune Gustavsson

Blekinge Institute of Technology, School of Engineering
P.o Box 520, SE37235 Ronneby, Sweden
Email: {christer.rindeback,rune.gustavsson}@bth.se

Abstract. Designing trustworthy e-services is a challenge currently undertaken by many actors concerned with the development of online applications. Many problems have been identified but a unified approach towards the process of engineering trustworthy e-services doesn't yet exist. This paper introduces a principled approach to deal with trust solutions in e-services based on a concern-oriented approach where end users' concerns serve as the starting point for the process to engineer appropriate solutions to trust related issues for an e-service. The trust management cycle is introduced and described in detail. We use an online application for reporting gas prices as validation of the proposed cycle.

Keywords: Trust, trustworthiness, trust management, e-services

1 Introduction

The dynamic human assessment of trust has been identified as a major concern for user acceptance and hence for deployment of efficient and successful online applications. Not only do we need to ensure trust, we need to create an environment and online support between end-users and e-service providers and other actors. This means to both engender, and provide means for a positive trusting experience. Many applications available online presuppose a high level of trust, and thus low levels of perceived concerns or risk taking by the user in order for users to utilize the provided service. Examples today include online-banking where sensitive financial information is exchanged between the bank and the bank customer on a public network (i.e. the Internet) or e-health applications where sensitive personal information might be sent over the public Internet infrastructure. It is interesting to note that use of online banking, when it was introduced in a larger scale a decade ago, indeed was regarded with scepticism by a substantial number of new users, but is today generally accepted as a trusted service. This example illustrates that trust in e-services is in fact dynamic in nature, depending on, as we will discuss later, time and context dependent variances in concerns and familiarity. In this paper we will address a principled way towards design, implementation, and maintenance of trustworthy e-services. To that end we will, in the next Section 2 Background, introduce a structured approach of addressing trust concerns of users and transforming those concerns into engineering principles of trustworthy systems. In

Section 3, we will introduce a trust management model geared at maintaining trustworthiness. In section 4, Validation, we analyse experiences gained from a field experiment where the e-service provided is ”Cheapest gas in the neighborhood!” [1]. In Section 5, Trust and trustworthiness, we summarize the main aspects on those topics. Thereafter, in section 6 Related work, we outline some other contemporary approaches towards trust and e-services. In section 7 Conclusions and future work we present our findings and point at future work. The final section is the Reference section.

2 Background

With a multitude of novel and existing e-services online questions and concerns regarding trust and credibility are of major concerns by stakeholders. In the ”early” days of e-commerce concerns about online payments was frequently discussed as a main barrier for successful online businesses [1]. Today general concerns about online payments are less prevalent and it is likely that issues related to this has been more grounded in general payment structures in our society, that is users won't generally have the same doubts with respect to online payments due to enforced and developed practices in the credit card payment industry.

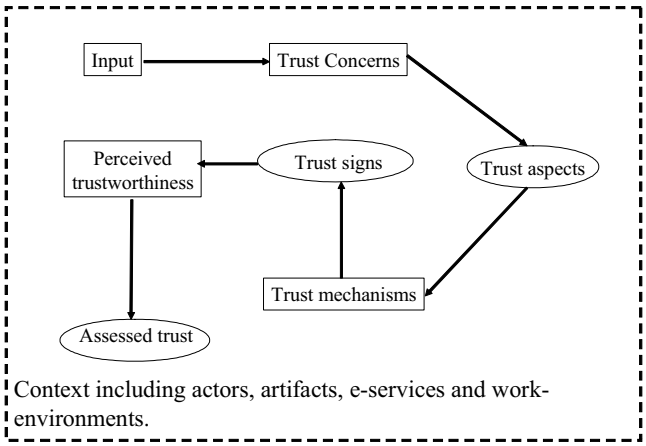
Privacy- and credibility concerns and qualities related to accuracy of information published on the Internet are other factors often mentioned in trust studies and literature. We will likely discover new concerns requiring attention in the future both in existing and new e-services. We claim in the paper ”Why trust is Hard” [2] that trust in artifacts is in fact an assessment by a user if a product or service is trustworthy. Building trustworthy systems is furthermore an engineering task based on a basis of observations about what is perceived to be trusting qualities. To support that task we have introduced a model that takes user related trust concerns and translate those into a set of trust aspects (e.g., my credit card number, or my identity can be stolen). For a given trust aspect there are usually several trust mechanisms that could be implemented to carter for the aspects (e.g., encryption of data or access control). Those mechanisms typically are invisible or difficult to assess by a common user. To that end, the service provider has to provide the service or product with signs (brand names, test results, and so on) to help the user to assess if the service meet her concerns in a way that ensure that the product is trustworthy. Trustworthiness, however, is a dynamic and context dependent concept since the perception of signs and the required mechanisms changes over time. In order to maintain trustworthiness we propose a trust management cycle (see fig. 1(b)). We will validate our model by applying it on the ”Cheapest gas in the neighborhood!” scenario example. As e-service designers we can attempt to find a balanced solution from a trust perspective that addresses the involved actors' different perspectives and trust concerns and turn them into appropriate mechanisms. Ideally the mechanisms should provide actors and end users with signs for trust assessment purposes. The relationships between these concepts are presented in fig.1(a)

[1] A Swedish web site: http://www.bensinpris.se. One of the authors is involved in the development of the site

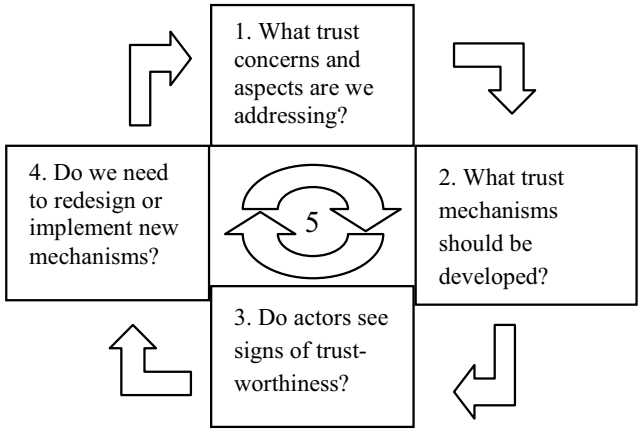
3 the Trust Management Cycle

In this section we will take a closer look at how our suggested approach can be broken down in a number of steps. In fig. 1(b) we introduce the trust management cycle, a model we propose to be used in order to develop sustainable trustworthy e-services.
*Step one: From Concerns to Aspects.*The first step of the trust management cycle is the initial component of our model. The dynamic nature of both concerns and perception of signs are forcing us to reflect and constantly re-assess our efforts to comply with the e-service users' concerns. We need to ensure that the identified trust issues are addressed. In fig. 1(c) we illustrate the dynamics of trust concerns.

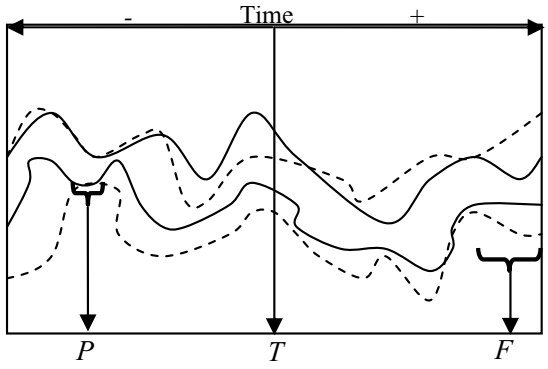
The gap between the upper and lower dotted curve illustrates the current set of addressed trust concerns of an e-service at a particular point in time. The filled curves illustrate the set of actual trust concerns at a point in time. With actual concerns we mean the set of the actors' and end-users' concerns with respect to a particular e-service at a particular time. We can illustrate points before T (e.g. P) and points in time after T (e.g. F). At time T in fig. 1(c) we can see that there is a mismatch between the current actual trust concerns and the addressed ones that may cause an insufficient treatment of relevant concerns for trust assessment purposes. There are other notable areas in fig. 1(c) of interest; at the time interval P we see an illustration of what we ideally want to achieve, namely a match between the actual concerns that needs attention and the identified concerns. Under such circumstances we are addressing all relevant trust concerns for the current context with respect to the e-service. At the point F we are addressing too many concerns that are outside the scoop of the actual concern domain at the time of investigation. This means that we may address issues that aren't actually perceived as a concern from the perspective of the involved actors and end-users. This can in turn lead to new concerns, e.g. if privacy is addressed in a context where it isn't perceived as motivated it may trigger trust concerns. Many concerns are similar and can be condensed into more general classifications. For instance if we have discovered many concerns related to the privacy of personal information this can be derived into the aspect privacy. We summarize this first step as: "What trust concerns and aspects are we addressing?".
Step two: from aspects to mechanisms. We can't deploy an aspect or trust concern directly into an e-service application or it's context. We need to make a transition between the identified concerns and aspects into deployable solutions. This process include efforts to discover, engineer and develop appropriate trust mechanisms. A trust mechanism is a deployable solution that encompasses one or more trust aspects. There is no one-to-one matching between a trust aspect and a trust mechanism; instead we may need multiple mechanisms to meet the concerns addressed. Consider for instance the case where the trust aspect privacy needs attention. First we may need to deploy a privacy policy but this may not be sufficient; we may also need to deploy and join a certification program such as truste in order to satisfy users' concerns for trust cues. A mechanism can also be used to satisfy multiple aspects. Such an example includes the mechanism "data encryption" which can be used to satisfy both security- and privacy aspects. The transition from concerns and aspects into the mechanism level is an important step since it is here decided what we will implement into the e-service with respect to trust. The mechanisms can be implemented in more than just one way. A privacy



(a) Important trust concepts



(b) Trust management cycle



(c) The dynamics of trust concerns

Fig. 1. The trust management cycle and the dynamics of trust concerns illustrated.

policy can, be a couple of rows long or span over several web pages. Thus each mechanism implementations may contribute uniquely to the addressed concerns and aspects. We conclude this step as ”What trust mechanisms should be designed to present signs related to the trust concerns and aspects?”

Step three: From mechanisms to signs. With deployed and implemented trust mechanisms in place the question is if our efforts are perceived by the actors and end-users for the cause of trust assessment. We can through our implemented solutions suggest that we are trustworthy and taking care of the concerns the end-users or actors may have. However trust and what the observers see is subjective; we can't enforce trust or the wished behavior onto the e-service end-users. We need to use measures to find out how and in what way our efforts are perceived by actors and end-users. Finding appropriate variables and measures is challenging and can be done by various means. Our trust management cycle in no way limits the approach to determine how the efforts made to encompass the identified trust concerns are made; however our approach is based on theories of signs, a concept used by [3] in the context of trust to illustrate the point that trust as such isn't a directly observable property, but rather we see signs suggesting trustworthiness such as ”an honest look” or affiliation. These signs are derived from the various observable properties in place (an honest person is a subjective observation based on e.g. the look of that person). Likewise a person may comment that somebody looks dishonest or unprofessional through his or her way of dressing.

The concept of signs in our context is linked to the deployed mechanisms implemented. Will there be signs pointing in the direction that the e-service provider seems to be ”honest” (a trust aspect we strive to encompass through mechanisms) or are signs and thus sources to assess the trustworthiness lacking? If we find a mapping between the identified concerns and the signs presented this is a step into a better-engineered e-service from the perspective of trust. The third step of the trust management cycle can be concluded as: ”Do actors and end-users see signs of trustworthiness that covers the identified concerns?”

Step four: Assessment. In the previous stage we discussed the relationship between signs, mechanisms and concerns. The re-assessment phase is the stage where we closer reflect upon the e-service context and the concerns we need to address in our solution. In fig.1(c) we illustrate the dynamics of concerns and we may need to reconsider from time to time which concerns we need to address in our solution. For instance at one point in time privacy may not be an issue but because of changes in attitudes in society and technical advancements these issues can become more important at certain points in time. This enforces us to assess trust issues during the life span of an e-service with arbitrary intervals. We also need to reflect upon the validity of the deployed trust mechanisms; are they still up to date or do they need fine-tuning? Should a particular mechanism be removed? If trust concerns are still in place we may need to introduce new mechanisms or re-design present ones for that partciular concern. In some cases e.g. awareness of legal changes or if a particular encryption chosen is hacked we may need to completely replace a mechanism. This step can be summarized as: ”Do we need to redesign or implement new mechanisms or address new concerns?”.

Step Five: Gateway to the next cycle. The concerns gathered and findings pointing us

towards assumptions that the trust concerns aren't properly addressed trigger a new round in the model. Reasons for this could be required changes of mechanisms or the introduction of new ones for a particular set of concerns. We also must consider if the concerns we are addressing are the right ones at this point in time.

4 Validation of our Model

Our "Cheapest gas in the neighborhood!" service will be used to illustrate how the trust management cycle can be used to reason about trustworthy e-service development. When the service was introduced the prices of gas in Sweden were varying almost daily due to the volatile global oil prices. Also there were local price wars where the price levels between different gas stations, could vary up to as much as 25% between closely located stations. Therefore people consult the service from time to time to find information about potential bargains on gas. If a price is lower in one station some may consider reschedule their route and in some cases even their destinations in order to save some money. The e-service relies on price reports submitted by it's visitors who report prices by filling out a form.

Concern: Price and information accuracy		
Cycle:	Mechanism:	Assessment of reactions:
1	Date policy published on site. Dates when the prices where reported added.	Ok but still comments where visitors want more information.
1	Forum and e-mail for feedback was deployed to enable comm-unication	A way to give feedback about site features.
2	Limits for false pricing was introduced.	Invisible feature as long as a user don't attempt to insert an unreasonable high or low price.
3	Registration requirements	Ok, but concerns raised regarding privacy.
4	Registration re-quirement became optional. Manual approval of ano-nymous reports.	The function opened up for anonymous submissions. Affected the will-ingness to report prices

Table 1. Condensation of mechanisms and assessments from our scenario.

When the service was first deployed only one concern had been identified on behalf of the end-users; the information about the prices needed to be correct and up to date. To avoid too old price quotes to be published a date limit mechanism was deployed. It was thought that this effort would mediate the correct signs for information accuracy. This concern seemed to be met during the initial stage of the e-service after deployment, but after a while it seemed like new mechanisms was needed in order to assure that the concern could be met.

During the first assessment cycle it turned out that users again were concerned about the accuracy the published prices. E-mails and online forum discussions revealed problems with false reports. The initial idea was that users would be able to spontaneously report price quotes without the need to provide an identity. By filling out a form on a web page the price report was published on the web site. After misuse, which affected end-users credibility in the e-service, mechanisms needed to be deployed in order to sustain the trustworthiness. The problem was addressed by forcing users to register in order to report prices by stating name and email. However, it turned out that some users where reluctant to sign up due to privacy concerns. For this reason the registration solution became optional with manual reviews of anonymous price quotes. Up until today this solution has proved to function well. In table 1 we illustrate findings from four cycles of the trust management cycle.

5 Trust and Trustworthiness

Trust can be defined in many ways depending on the context and circumstances, there simply is no commonly agreed upon definition stating what trust actually is. Trust is often seen as a mechanism used to reduce complexity in situations of uncertainty [4]. If an online service is perceived as to be trusted this may increase the likelihood that the service is used by the truster although this is no guarantee. Trust is by no means static, we may have trust in e.g. a neighbor but due to some event that ends with disappointments about their behavior, or solely a suspicion about a behavior, may cause the trust to decrease. The very opposite may also be true since trust can grow depending on signs suggesting somebody is to be trusted. We have identified the following trust dimensions [2]:

Trust in Professional Competence - When a decision to delegate a task to another actor is taken this is often based on a perception of that actor's professional competence. This refers to expectations about the professional abilities [5] of e.g. a doctor or banker and suggests further refinements of trust expectations.

Trust in Ethical/moral Behavior - Trust isn't only related to professionalism in dealing with tasks as such, it is also suggested to be linked to values and less tangible nuances such as ethical and moral premises. If a trusted professional acts in a manner that is perceived as being against common ethical and moral norms we can choose to distrust this person in a given context despite his professional skills. Examples include certain types of medical experiments or other acts that can be regarded as unethical or even criminal if detected. Trust in moral or ethical behavior is, of course, very context de-

pendent. Moral and ethical trust is discussed both in [5] and [6].

Trust in Action Fulfillment - In cooperation a specific trust dimension surfaces in most contexts. That is, can a subject trust that an object will indeed fulfill a promise or obligation to do a specified action? When ordering a product online concerns may for instance be raised if it will be delivered or not.

Functionality - The functionality of an artifact is an important and natural quality of trust, e.g., the tools are expected to function as they should. An implicit trust condition is that an artifact or tool is not behaving in an unexpected or undesired way by its design [7].

Reliability - The reliability of an artifact is another important criteria of trust in classical artifacts. The tools should be resistant to tear and wear in a reasonable way and the e.g. a VCR should function flawless for some years. Reliability thus means that an artifact can be expected to function according to the presented functionality and is working when needed.

These dimensions of trust are related to one or more objects, in which a truster place his or her trust [2]:

Trust in social/natural order and confidence - Our society rests on basic assumptions about what will and will not happen in most situations. For instance we have trust in the natural order, that the heaven won't fall down or that the natural laws will cease to be true. There is also a general trust related to the social order in most of our societies, that is that the governmental representatives will do the best for the citizens and countries they represent and follow laws and norms as well as follow established practices accordingly. This mutual trust isn't something that actors in general reflect consciously about. The non-reflective trust serves as a basic trust/confidence level for our daily actions where in general there aren't any alternatives to the anticipated risks. The notion confidence [4] is sometimes used in situations where actors in reality have no choice. It isn't a viable option to stay in bed all day due to concerns about the social or natural order.

Trust in communities - Humans are often part of a larger community. In the society we have companies, non-profit organizations, governmental institutions and other groups of humans, which often act according to policies, and interests of the community. In many cases the trust may be attributed primarily (or at least in part) in the behavior in a community e.g. a hospital. On the other hand, a hospital may be perceived as trustworthier than another due to better reputation regarding the perceived treatment and quality of their staff. Depending on the context, trust by a subject may be placed on the object being a community, an individual representing the community, or both.

Trust in humans - In many situations we attribute trust towards other humans, we may trust a particular person about his capabilities or trust his intentions about a particular action. When buying a used car for instance we may trust a car salesman to a certain degree or trust a neighbor being an honest person. Trust between humans has been stud-

ied among others by [8–10].

Trust in artifacts - Trust in human-made objects such as cars or VCR:s are in some cases discussed in a manner which implies that these objects can be seen as objects in which trust is placed. For instance 'I trust my car and VCR'. This means that our expectations regarding the objects with respect to reliability are in some sense confused with or attributed for trust in humans enabling the intended behavior.
The distinction between what or whom we trust is important when determining how to address trust issues in a given context. E.g. if a patient don't trust a doctor this may have different causes which can be related to the person as such (a human) and his or her professional competence, artifacts or maybe the reasons are that the hospital as such isn't to be trusted, the doctor is just the representative towards the lack of trust is attributed. When developing e-services we also must consider the cause of particular trust concerns. Are for instance the concerns technology-oriented or are there other reasons for the identified trust concerns?

Trust and trustworthiness shouldn't be confused. An online vendor or a person can suggest that they are trustworthy by their actions or through e.g. a web site [11]. But it is up to the truster to assess these suggestions.

6 Related Work

Two strands of dealing with the problem of trust have been identified. In some work and traditions there is an implied idea that we can solve the problems related to trust by encryption and security solutions [1]. Others suggest the need to create an atmosphere of trust and understand issues related to different situations and actors. One approach that deals with trust in risky environments such as e-commerce is the model of trust in electronic commerce (MoTech) [12]. It aims to explain the factors that affect a person's judgment of an e-commerce site's trustworthiness. MoTech contains of a number of dimensions intended to reflect the stages visitors goes through when exploring an e-commerce website. The dimensions pre-interactional filter, interface properties, informational content and relationship management will be described below. Each of these components addresses factors that have been observed to affect consumers' judgment of an on-line vendor's trustworthiness. Pre-interactional filters refer to factors that can affect people's perceptions before an e-commerce system has been accessed for the first time. The factors presented are related to user psychology and pre-purchase knowledge. The first group refers to factors such as propensity to trust and trust towards IT in general and the Internet. Pre-purchase knowledge is related to Reputation of the industry, company and Transference (off-line and on-line). The second dimension of MoTech is concerned with interface properties that affect the perception of a website. Here the components are branding and usability. Factors in the branding component are appeal and professionalism. The usability component factors are organization of content, navigation, relevance and reliability. The next dimension, informational content contains components related to competence of the company and the products and services offered and issues regarding security and privacy. The fourth and last dimension reflects the facilitating effect of relevant and personalized vendor-buyer relationship. The com-

ponents Pre-purchase Interactions and Post-purchase interactions are related to factors such as responsiveness, quality of help and fulfilment. This model is interesting and divides the relationship between the vendor and user into units that can be analysed further.

7 Conclusions and Further Work

We introduced the trust management cycle, a principled approach to address the problem of trust in online e-services in a structured manner which takes a starting point in actual trust concerns expressed or identified in the user community. These concerns need to be turned into deployable solutions by means of trust mechanisms. We also presented validation of the cycle by applying it on the ”cheapest gas in the neighborhood!” scenario. We need to further validate the model and investigate constituents of e-service contexts in order to find better ways to deal with trust issues. We also see a need to understand the relationship between specific signs and mechanisms in order to better understand the characteristics of good trust mechanisms. This will hopefully give us better tools to deploy and design trustworthy e-services.

References

1. Nissenbaum, H.F.: Can trust be secured online? a theoretical perspective. In: A Free Information Ecology in the Digital Environment, New York, NY (2000)
2. Rindebäck, C., Gustavsson, R.: Why trust is hard - challenges in e-mediated services. In Falcone, R., Barber, S., Sabater-Mir, J., Munindar, S., eds.: Trusting Agents for Trusting Electronic Societies: Theory and Applications in Hci and E-Commerce. Volume 3577. Springer Berlin / Heidelberg (2005) 180–199
3. Bacharach, M., Gambetta, D.: Trust as type detection. In Castelfranchi, C., Tan, Y.H., eds.: Trust and deception in virtual societies. Kluwer Academic Publishers, North Holland (2001)
4. Luhmann, N.: Familiarity, confidence, trust: Problems and alternatives. In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. Basil Blackwell, New York, NY (1988) 94–110
5. Barber, B.: The logic and limits of trust. Rutgers University Press, New Brunswick, N.J. (1983)
6. Baier, A.: Trust and antitrust. Ethics **96** (1986) 231–260
7. Muir, B.M.: Trust in automation .1. theoretical issues in the study of trust and human intervention in automated systems. Ergonomics **37** (1994) 1905–1922
8. Deutsch, M.: The resolution of conflict; constructive and destructive processes. Yale University Press, New Haven,NY. (1973)
9. Gambetta, D.: Can we trust trust? In Gambetta, D., ed.: Trust : Making and Breaking Cooperative Relations. B. Blackwell, New York (1988)
10. Rempel, J., Holmes, J., Zanna, M.: Trust in close relationships. Journal of Personality and Social Psychology **49** (1985) 95–112
11. Sisson, D.: e-commerce: Trust & trustworthiness (2000)
12. Egger, F.N.: From Interactions to Transactions: Designing the Trust Experience for Business-to-Consumer Electronic Commerce. PhD thesis, Technische Universiteit Eindhoven (2003)

Paper 4: Functional versus Non-Functional Requirements Considered Harmful

Gustavsson, R., J. Lundberg, C. Rindebäck, and K. Ådahl. (2007)
In the proceedings of the 2007 World Congress in Computer Science, Computer Engineering, & Applied Computing, Las Vegas, NV.

Functional versus non-functional requirements considered harmful

Rune Gustavsson, Jenny Lundberg, Christer Rindebäck and Kerstin Ådahl

Blekinge Institute of Technology, School of Engineering
P.o Box 520, SE37235 Ronneby, Sweden
Email: {rgu,jlu,cri,kad}@bth.se

Abstract. Requirement elicitation and requirement engineering are crucial steps in system design, implementation, and maintenance. In addressing software intensive distributed socio-technical system such as trusted workflows in life-critical situations or information sharing support systems, we claim that the traditional division between functional and non-functional requirements is difficult and even harmful to maintain. We propose in this paper a more principled and context dependant approach towards requirement engineering and high-level system validation.

Keywords: Requirement elicitation, Requirement engineering, non-functional requirements

1 Introduction

We have during the last decade designed and implemented software intensive systems in the areas of distributed healthcare [1–3], critical infrastructures [4–9], workflow support in lifecritical situations [10–12], and information sharing [13]. Lessons learned from those investigations are that a principled approach of requirement elicitation and requirement engineering is an important ingredient in designing, implementing, and maintaining trustworthy socio-technical systems. As a matter of fact, the most crucial requirements underpinning user acceptance and trust falls within what traditionally is termed non-functional requirements. A simple example, stakeholders involved in system specifications might have concerns related to information security (can be classified as a non-functional requirement). However, a mean to implement information security is to apply cryptography (a functional transformation of information). This coding is typically a result of applying an algorithm. That is, a nonfunctional requirement has in the design and implementation phase been transformed into a function (algorithm) that in turn has to compete with other algorithms for resources at runtime. Another important lesson that follows from this typical example is that we cannot focus on ”an important class of functional requirements” when designing and implementing systems and hope to add other requirements as add-ons. All system-sensitive requirements have to be addressed properly from the outset of the system engineering efforts. We argue that the present classification of requirements into functional or non-functional requirements has at least the following shortcomings:

1. Presupposes a fixed ontology of functional and non-functional requirements independent of stakeholders and context.
2. Might focus on non-relevant issues found out at later stages of the system development process.
3. Does not easily allow assessments of trade-offs between requirements.

These shortcomings might lead to focusing on wrong and/or non-important requirements while not detecting important requirements when designing and implementing *trustworthy software-intensive systems*. In short, we regard the current principle of distinguishing between functional and non-functional requirements as harmful much in the spirit of Dijkstras remarks on the harmful effects of using the GO TO statements in programming [14]. He argued that this "natural" programming style in fact produced programs that were difficult to validate and maintain. To remedy a similar perceived concern related to requirement engineering of systems we propose a *configurable methodology* supporting requirement elicitation and engineering based on a *clear ontology* and addressing related *epistemological issues* in a principled way. The remaining part of the paper is organized as follows. In *Section 2 Background*, we give a personal view of issues related to the concepts of functional and non-functional requirements including motivations and issues brought forward in literature. The section ends with a list of shortcomings related to the dichotomy of requirements. Those shortcomings are the starting points of our presentation in *Section 3 Our approach towards quality assured system development*. The paper ends with *Section 4 Conclusions and further work* and *References*.

2 Background

The mathematical model of computing is the *Turing machine* from 1936 [15], later implemented using the *von Neumann architecture* that is mainly with us today [16]. That is, a computation is modeled as:

Computation = Algorithm + Abstract machine (1)

Equation (1) explicitly states the ontology related to computation. That is, Computation, Algorithm, Abstract machine. A natural set of epistemological issues is Development and performance of algorithms, Relations between Algorithms and Abstract machine. The von Neumann architecture of the *programmable Turing Machine* introduced the *implementation ontology* of Memory, Control unit, Arithmetic Logic Unit, Input, Output and corresponding epistemological issues. The algorithms were later implemented using suitable programming languages that could be interpreted or compiled into machine executable code. In fact, during the decades 1940's to 1960's the main efforts in computer science and later software engineering were devoted to defining suitable languages, programming methods and corresponding abstract machines and compilers for new classes of computations. The predominant classes of computations in the beginning were scientific computations (assembler, Fortran) and basic administrative computations (COBOL). The main difficulties addressed were to develop, implement, and maintain *correct* and *efficient algorithms* for *batch computing* in well understood

application domains. That is, requirements related to the (implicit) ontology related to equation (1). *Correctness* of algorithms (doing the right thing) is of course an important requirement (later a functional requirement). At the same time efficiency (*performance*) is an additional requirement (sometimes later called a non-functional requirement) that could and should be *measured*.

During the 1960's we learned how to utilize trade-offs between algorithms and data structures to improve software engineering (doing the things right) as well as performance for certain classes of computations, e.g., real-time critical computations or data base applications. During the 1970's new kinds of computations appeared such as distributed computing (clientserver models), networking, and standalone knowledge intensive expert systems. In short new types of system requirements surfaced due to new kind of applications (not so well understood) and/or new models ofconnectivity where new kinds of end-users could get access to computing power (the PC and Macintosh). Needless to say, the early rather clear-cut division between functional and non-functional requirements became more blurred and several attempts were made to clarify the concepts. In fact as computation (programs) become embedded in applications serving a broader class of users new user related, system related, and business related requirements were added in a rather ad-hoc way. In principle the computational model, or ontology, behind (1) was not explicitly changed reflecting the new kinds of computations.

The concepts of Users, Markets, Systems are evidently not parts of (1) nor corresponding epistemological issues from which we can derive models supporting, e. g., requirement engineering in a principled way. A standard book on requirement engineering is Kotonya - Sommervilles *Requirements Engineering Processes and Techniques* [17] from 1998. Recent books and papers addressing requirements are *Mastering the requirements process* [18] from 2006, Software Requirements: Styles & Techniques [19] from 2002, and *Modeling Architectural Non Functional Requirements: From Use Case to Control Case* [20] from 2006. In the first book there is still a sharp division between functional and non-functional requirements whence in the second book the author use the labeling of *quality attributes* to cover a broader spectrum of requirements. In this book there is also a more comprehensive view on requirement elicitation and on stakeholders. The paper that is specifically addressing non-functional requirements is focusing specifically on Use Cases to that end. In short, system requirement elicitation and engineering has during the last decades since long in practice surpassed the ontology focus of equation (1) above. In a paper [21] Urrego-Giraldo's acknowledges the considerable work done in the field of Non-Functional requirements in the last ten years. Urrego-Giraldo's own contribution to this discussion is an approach in the same direction as we propose, that is, aiming to integrate functional and non-functional requirements in the same analysis process. Similar attempts can be found in [22] where the authors argue that non-functional requirements is necessary to implement in the design process as early as possible, and in [23] where means of managing the impact on functional changes upon non-functional requirements are presented. There are four principal views of requirements, corresponding to the natural stakeholders *developers, users,* and *owners*, that is:

1. The (software) product view: Requirements related to the services that the product should achieve.
2. The (software) process view: Requirements on the process to achieve the proper product.
3. The end-user view: Requirements from the user to achieve user acceptance.
4. The product owner view: Requirements related to product business processes.

Furthermore, we have stakeholders representing *regulators* and *infrastructure owners*. Typically those stakeholders provide *constraints* on system use and performance. In the following we will focus on the first group of stakeholders. We argue that present day classifications of functional and non-functional requirements are not derived from concerns by stakeholders, but from rather ad-hoc assumptions that often mixes the concerns of the (implicit) stakeholders. The now classical division in functional and non-functional requirements has several shortcomings as a basis for a configurable and comprehensive methodology supporting requirement engineering of future *software intensive and trustworthy distributed systems*. A short list:

1. Ad hoc classification principles manifested by different types of classification schemas.
2. Implicit ordering of importance. Typically a ”non-requirement” is of lesser importance than the basic ”requirement”.
3. New requirements might appear late in the system development cycle.
4. Difficulties to assess trade-offs between requirements.
5. Difficulties to address proper validations or assessments of requirements.
6. Focus on validations of requirement documents rather than assessments on a systemic level.

In the next section we present our approach of requirements elicitation and engineering to remedy some of those shortcomings.

3 Our approach towards quality assured system developement

Our starting point for a principled approach of requirement acquisition and requirement engineering are identifying concerns by stakeholders. That is owners, developers, and users. The epistemological issues are related to which concerns we can gain knowledge about and validate of the socio-technical-economic system at hand. Ontological issues underpins the modeling we need to support requirement elicitation and requirement engineering. We propose a service-oriented view of socio-technological-economic systems [10, 24–26, 4, 27–29, 11, 13, 12]. A system provides a set of services that are used to carry out a (general) business purpose [1]. System components typically consist of hardware, software, data, and workers. Systems are specified by the services they provide along with other requirements such as reliability or low cost of ownership. A system design consists of specifying components, their attributes, and their relationships.

[1] Blanchart and Fabrycky, System Engineering and Analysis (Thrid Edition), Prentice Hall, 1998.

3.1 Ontological and epistemological frameworks

Our starting point is the CommonKADS configurable methodology that supports design and maintenance of knowledge intensive systems [30, 31]. The methodology focus on the type of applications we are targeting we can then configure the methodology accordingly. As a starting point of this configuration we decide about the relevant worldview, that is, the relevant epistemology and ontology for the system at hand. Epistemological issues have two dimensions. Internal epistemology related to what the system should be able to know about itself and/or its environment. External epistemology related to what a user and/or maintainer should know about the system. In short, the internal instrumentation determines the abilities of self* services, whence the external instrumentation determines the user views of the systems as well as the possibilities of the system administrator to control and support proper system behaviour. The targeting of the worldview towards the goal system (the "Common-KADS pyramid") includes choices of appropriate theories, methods, and tools to support the engineering of the system at hand. In fact, the Methodological pyramid of Figure 1 replaces equation (1) above as the notation of: software intensive socio-technological-economic systems as the goal system (2) The corresponding identification of ontologies and epistemic issues are parts of our configurable methodology (Section 3.2).

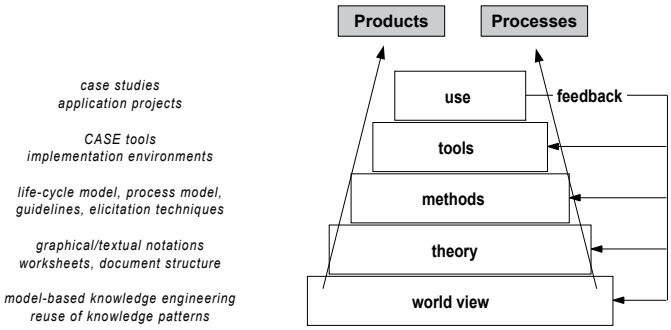


Fig. 1. The CommonKADS methodology pyramid. The main product components and processes of the methodology are indicated.

The CommonKADS methodology leaves open the implementation process as such but advocates implementations of rule bases since the original methodology aimed at stand alone, or well separated, knowledge systems. The use of a model based as well

as a risk driven approach to system development makes CommonKADS the methodology of choice (configurable and comprehensive) in development of knowledge intensive systems [32, 33, 31, 34]. The second source of out methodology is the IEEE standard 1471-2000[2] that addresses the activities of the creation, analysis, and sustainability of architectures of software-intensive systems, and the recording of such architectures in terms of architectural descriptions. The IEEE conceptual model includes: Mission, Environment, System, Architecture, Stakeholders, Architectural description, Rationale, Concerns, Viewpoint, View, Library viewpoint, and Model with relations. The standard enforces a principled way of designing suitable architectures.

3.2 A configurable methodology supporting requirement elicitation and engineering

The main activities in the initial phases of our methodology are (c.f., Figure 1): a) Identification of the goal system and worldview including ontologies. b) Identification of *stakeholders*, their roles and needed competences. c) Identifications of *concerns* by the stakeholders individually and *systemic* concerns. d) Identification of *epistemological* issues supporting analysis and modelling of concerns. e) Identification of *requirements* related to concerns as well as ontologies and epistemologies supporting elicitation, engineering, *assessments*, and *validation* of measures ensuring system behaviour respecting concerns. f) Instantiation of the methodology supporting system *requirements engineering*, design, implementation and maintenance based on the generalised CommonKADS methodology and the IEEE standard. We now shortly address issues related to identification of stakeholders concerns. We have developed a *trust model* to support design and maintenance of trustworthy systems, or rather, trustworthy computer artifact mediated services (e-services), [2, 3]. The model was developed and partly implemented in a project on trust in e-services in distributed health care[3] [1]. Our specific concern in that setting was trusted delegation of tasks and proper system support to that end. The following Figure 2 captures the main components of our trust model. The bottom line in our trust model is that potential users of the offered service express their trust concerns related to whether to use the service or not. Those concerns are then classified into different information types [35, 36]. A goal of designing and implementing trustworthy systems is to identify and implement mechanisms that can be communicated to the user via signs in a way that reflects her specific concerns regarding using the service at hand, that is, assessments whether or not the stakeholders concerns have been adequately addressed. In all our applications several trust concerns related to information management have emerged. Typically, those trust concerns addresses one or more of the following issues: a) Adequacy and integrity of information. Do we have the information we need and can we trust it? b) Non-disclosure of information. Is the information protected from non-intended use? c) Availability of information. Will I get the right

[2] IEEE Recommended Practice for Architectural Description of Software-Intensive Systems, IEE Std 1471-2000.

[3] The EU funded project Alfebiite - A logical framework for ethical behaviour between infohabitants in the information trading economy of the universal information ecosystem. IST-1999-10298.

information at the right time? d) Common situation awareness. Have we all the same situation assessment for appropriate actions to fulfill the mission? e) Breakdowns. What happens if something goes wrong? Can I be accountable or liable? f) Traceability. Can my actions be recorded by the system? A blessing and a threat! The first three concerns a commonly referred to as the CIA (Confidentiality - Integrity - Availability) concerns of information security and is in focus of information security engineering. However most of those contemporary efforts and solutions are context *independent* whence we emphasise the importance of the mission context to allow for customized solutions [29, 5, 8, 9]. The last three trust concerns mentioned above have, however, very little attention in R&D efforts at the moment. This is partly due to fact that the rapid emergency of new application areas, based on computer-enabled interaction between groups in missions, has not a clear research agenda at the moment. The purpose of this paper is to outline some important steps to that end. Since trustworthiness is a *descision by the user*, based on assessing relevant information, a challenge is to convey the information that the system indeed meets the concerns articulated by the user by communicating appropriate signs conveying the right information [7, 9, 2, 3].

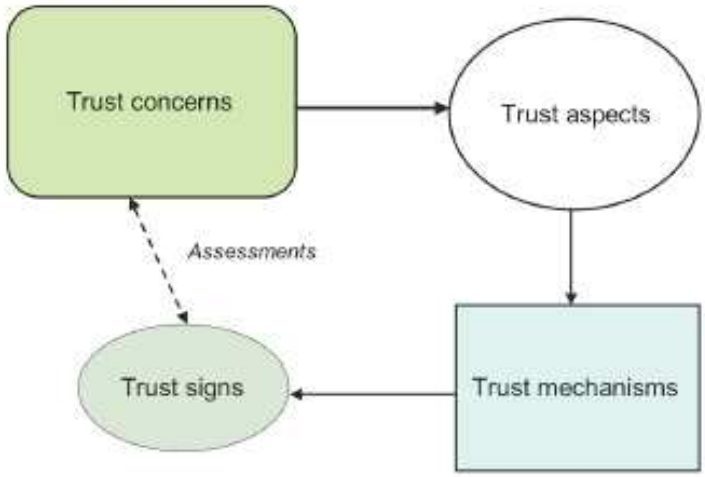


Fig. 2. A trust model highlighting the relations between trustworthiness assessment, trust concerns, trust mechanisms, and signs

Figures 2 and 3 highlights the importance of context dependant communication of information in designing and maintaining trustworthy systems, or rather services [35–37]. The outcome of the assessment clearly depends on the context in which the service will be used and how well the interpretation of signs meets the trust concerns. We thus have to investigate the anatomy of information, language and communication in order to design and maintain trustworthy information management agents. These issues have

been addressed in several recent thesis works published by our research group [10, 38, 11, 7, 13, 12].

3.3 Requirements engineering of information centric critical systems

The main activities of our *requirements elicitation and engineering methodology* are as follows:

1. Identification of work practices
2. Identification of workflows
3. Identification of supporting services to workflows
4. Configuration of workflow services and supporting middleware services in a Service Oriented Architecture
5. Configuration of Service Level Agreements to support coordinate and to monitor execution of services
6. Configuration of mission oriented applications

Figure 3 illustrates the use of Environment models (organisation, tasks, agents) of workflows and the mapping (Model Based Design) onto the SOA architecture. The Coordination model and Communication model constitute the interfaces (protocols and data models) to be agreed upon and maintained by the SLAs (Service Level Agreements). The output from the activities is: (1) Identification of work practices and (2) Identification of workflows are input to activities related to Figure 3. That is, (3) Identification of supporting services to workflows, (4) Configuration of workflow services and supporting middle-ware services in a Service Oriented Architecture, and (5) Configuration of Service Level Agreements to support coordination, monitoring, and execution of services.

he set of models supporting high-level requirement engineering of services and corresponding SLAs supports Model Driven Development (MDD) of software systems [33]. Similar ideas can also be found in efforts on Meta-Model based approaches to information system engineering [34]. Service level agreements are in focus of several ongoing Grid projects. A generic reference is CoreGrid[4] . Complementary work based on web services are conducted by the W3C consortium specifically on web services activities and coordination[5] . The workflow support is mostly focused on issues related to Business Process Execution Language for web services (BPEL4WS). In our approach we have to address a richer SLA context (Figure 3) than the Grid or W3C efforts that focus more on a bottom-up approach. Furthermore, agent enabled coordination enables also a more flexible and intelligent SLA management, e.g., supporting self-healing of missions [27]. We have focused our R&D on dependable and secure socio-technical systems (information-centric MAS) on protecting system, and specifically software, execution. To that end we have developed tools and methods of experimental online engineering supporting design, implementation and maintenance of dependable systems. There are four different types of interfaces to the system [38, 25, 5, 7–9]:

[4] www.coregrid.net/
[5] www.w3.ws/2002/ws/cg/

Fig. 3. A revised CommonKADS model suit supporting modelling of workflow services and interface to Service Level Agreements

1. Support for experiment set up, execution, monitoring, and evaluation
2. Support for configuration and maintenance of missions
3. Support and monitoring of the entire system
4. Support for different views of the system by different users and user groups

In our methodology the mission is first modeled and partially implemented articulating the dependability issues at hand. The relevant inspection points supporting monitoring of dependability aspects are identified and properly instrumented. The behavior of the system is observed and assessed. Proper steps for the next development cycle are identified (risk based evaluation) constitute the requirements for the next cycle.

3.4 Validations of requirements

We are addressing different views and concerns on socio-economic-technical systems exemplified in Section 1. The concerns related to the socio-aspects can be summarize as trustworthy support of workflows, whence the economic concerns are focusing on the economy of service oriented business processes related to in- and outsourcing of services and maintenance and new businesses. The concerns related to the technical systems are related to dependable and resilient system behavior. We have outlined how we can translate the different concerns into system requirements highlighted in Figures 2, and 3. Figure 3 depicts also the architecture for our validation experiments. In the experiments and validations different *operationalization of the SLA component* of Figure 3 are implemented. As we earlier have noted is the SLA component the bridge between the high-level workflow support and the low-level system services. In effect we have the following main types of validations:

1. Validation of requirement engineering. Are the concerns related to workflows properly captured and captured into SLA and supporting services?

2. Validation of business processes. What is the core services underpinning the business processes? What are the trade-off between in- outsourcing of services in maintenance our creating new business processes (new services)?
3. Validation of dependable and resilient system behavior of software intensive systems!

Validations of the approach outlined in this paper are addressed in the thesis works of Patrik Brandt, Jenny Lundberg, Per Mellstrand, Christer Rindebäck, and Louise Östlund [10, 11, 7, 3, 12]. Validations of proposed workflows against ethnographical findings are specifically addressed in [11]. Identification of workflows at new Call Centres is particularly addressed by [10, 12], The role of situation semantics and mapping on services are in focus of [10], whence aspects of core business services and issues of in-sourcing and out-sourcing and maintaining team competence are in focus of [12]. Models and techniques supporting design and maintenance of trustworthy services are in focus of [3]. Experimental support and methodologies supporting validation of trustworthy execution environments of software intensive systems is addressed in [7].

4 Conclusions and further work

We have presented a configurable methodology supporting the whole life cycle of dependable network enabled systems. We have specifically addressed requirement elicitation and requirement engineering for software intensive information centric systems supporting trustworthy workflows in critical situations. The methodology is supported by a set of models and tools as well as environments supporting simulations and experiments. The architecture of the methodology can be expressed using IEEE standards. Furthermore our methodology allows imports of technologies such as agent technologies at appropriate places of the system life cycle. In the paper we have focused on identified shortcomings of present day requirement engineering models, that is requirement engineering and implementation of trustworthy service oriented systems. Our configurable methodological approach involves proper choices of methods, models and techniques as exemplified in the paper. The models (meta models) are the results of the requirement process and input to the design and implementation phases. A key component is the Service Level Agreement (SLA). The SLA implements functional aspects of the requirements onto services but also provides means supporting non-functional aspects of the system. Agent technologies plays important roles at different stages of the development processes:

1. During requirement analysis the high-level multi agent modeling supports the ethnographical and information centric analysis of work processes.
2. The requirements and design of SLAs benefits from techniques and models from architectures and models from MAS, that is agent capabilities and coordination models.
3. Design of services again uses selected models and techniques from MAS.

The work reported in the paper is naturally on work in progress. Our focus is now on the requirement engineering processes and on validation challenges, not the least on trust aspects, semantics and self-healing.

References

1. Gustavsson, R., Fredriksson, M., Rindebäck, C.: Computational ecosystems in home healthcare. In Dellarocas, C., Conte, R., eds.: Social Order in Multiagent Systems. Volume 2 of Multiagent Systems, Artificial Societies, and Simulated Organizations. Kluwer Academic Publishers, Boston (2001) 201–220
2. Rindebäck, C., Gustavsson, R.: Why trust is hard - challenges in e-mediated services. In Falcone, R., Barber, S., Sabater-Mir, J., Munindar, S., eds.: Trusting Agents for Trusting Electronic Societies: Theory and Applications in Hci and E-Commerce. Volume 3577. Springer Berlin / Heidelberg (2005) 180–199
3. Rindebäck, C.: Designing and maintaining trustworthy online services. Licentiate thesis, School of Engineering, Blekinge Institute of Technology (2007)
4. Gustavsson, R.: Ensuring dependability in service oriented computing. In: Proceedings of The 2006 International Conference on Security & Management (SAM06) at The 2006 World Congress in Computer Science, Computer Engineering, and Applied Computing, Las Vegas. (2006)
5. Gustavsson, R., Mellstrand, P.: Ensuring quality of service in service oriented critical infrastructures. Journal of Critical Infrastructures (IJCIS) (2007)
6. Mellstrand, P.: Protecting software execution by dynamic environment hardening. Licentiate thesis, Blekinge Institute of Technology (2005)
7. Mellstrand, P.: Informed System Protection. PhD thesis, Blekinge Institute of Technology (2007)
8. Mellstrand, P., Gustavsson, R.: An experimental driven approach towards dependable and sustainable future energy systems. In: Proceedings of Third International Conference on Critical Infrastructures, Alexandria VA. (2006)
9. Mellstrand, P., Gustavsson, R.: Experiment based validation of ciip. In: Proceedings of 1st International Workshop on Information Infrastructure Security (CRITIS06), special track at 9th Information Security Conference (ISC 2006). (2006)
10. Brandt, P.: Information in use: Aspects of Information Quality in Workflows. PhD thesis, Blekinge Institute of Technology (2007) ISSN 1653-2090 ISBN 978-91-7295-111-2.
11. Lundberg, J.: Principles of workflow support in life critical situations. Licentiate thesis, School of Engineering, Blekinge Institute of Technology (2007)
12. Östlund, L.: Information in use: In- and Outsourcing Aspects of Digital Services. PhD thesis, Blekinge Institute of Technaology (2007) ISSN 1653-2090 ISBN 978-91-7295-110-5.
13. Adahl, K.: Transparency of critical information for patient empowerment in e-health. Licentiate thesis, School of Engineering, Blekinge Institute of Technology (2007)
14. Dijkstra, E.W.: Letters to the editor: go to statement considered harmful. Commun. ACM **11** (1968) 147–148
15. Turing, A.: On computational numbers, with an application to the entscheidungsproblem. In: Proceedings of the London Mathematical Society, Series 2, 42. (1938)
16. von Neumann, J.: First draft of a report on the edvac. Contract no. w-670-ord-4926, between the united states army ordnance department and the university of pennsylvania moore school of electrical engineering, University of Pennsylvania (1945)
17. Kotonya, G., Sommerville, I.: Requirements engineering : processes and techniques. Worldwide series in computer science. J. Wiley, Chichester ; New York (1998)
18. Robertson, S., Robertson, J.: Mastering the requirements process. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA (1999)
19. Lauesen, S.: Software requirements: styles and techniques. Addison-Wesley, Harlow (2002)
20. Zou, J., Pavlovski, C.J.: Modeling architectural non functional requirements: From use case to control case. icebe **0** (2006) 315–322

21. Urrego-Giraldo, G.: Reasoning nonfunctional goals and features in web systems. Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference on (2004) 643–644
22. Jonkers, H., Iacob, M.E., Lankhorst, M.M., Strating, P.: Integration and analysis of functional and non-functional aspects in model-driven e-service development. edoc **0** (2005) 229–238
23. Cleland-Huang, J., Settimi, R., BenKhadra, O., Berezhanskaya, E., Christina, S.: Goal-centric traceability for managing non-functional requirements. In: ICSE '05: Proceedings of the 27th international conference on Software engineering, New York, NY, USA, ACM Press (2005) 362–371
24. Canfora, G., Penta, M.D.: Testing services and service-centric systems: Challenges and opportunities. IT Professional **8** (2006) 10–17
25. Fredriksson, M., Gustavsson, R.: Online engineering and open computational systems a unified approach towards theory and practice of agent systems. In Bergenti, R., Gleizes, M.P., Zambonelli, F., eds.: Methodologies and Software Engineering for Agent Systems. Kluwer academic publishers (2004)
26. Fredriksson, M., Gustavsson, R.: A methodological perspective on engineering of agent societies. In: Engineering Societies in the Agents World II: Second International Workshop. (2004)
27. Gustavsson, R., Fredriksson, M.: Process algebras as support for sustainable systems of services. process algebras as support for sustainable systems of services. In Viroli, M., Omicini, A., eds.: Communication and Computing. Springer Verlag (2005) 179–203
28. Gustavsson, R., Fredriksson, M.: Sustainable information ecosystems. In Garcia, A., Lucena, C., Zambonelli, F., Omicini, A., Castro, J., eds.: Software engineering for large-scale multi-agent systems: Research issues and practical applications. Volume 2603 of Lecture Notes in computer science series (LNCS). Springer Verlag (2003) 127–142
29. Gustavsson, R., Mellstrand, P.: Ensuring quality of service in service oriented critical infrastructures. In: Proceedings of International Workshop on Complex Network and Infrastructure Protection. (2006)
30. Akkermans, J.M., Gustavsson, R., Ygge, F.: Structured engineering process for agent communication modelling. In: Knowledge Engineering and Agent Technology. IOS Press (2000)
31. Schreiber, G., Akkermans, H., Anjewierden, A., Hoog de, R., Shadbolt, N., Velde Van de, W., Wielinga, B.: Knowledge engineering and management : the CommonKADS methodology. MIT Press, Cambridge, Mass. (2000)
32. Becker, J., Dreiling, A., Ribbert, M.: Meta-model-based approaches to information systems engineering (2003)
33. Elvesaeter, B., Hahn, A., Berre, A.J., Neple, T.: Towards an interoperability framework for model-driven development of software systems. Technical report, Publications by EC funded Integrated Project ATHENA (2005)
34. Zinnikus, I., Benguria, G., Elvesaeter, B., Fischer, K., Vayssiere, J.: A model driven approach to agent-based service-oriented architectures. In: In Proceedings of the ATOP Workshop at AAMAS 2006. (2006)
35. Barwise, J., Perry, J.: Situations and Attitudes. Center for the Study of Language and Inf (1998)
36. Devlin, K.: Logic and Information (Cambridge Tracts in Theoretical Computer Science). Cambridge University Press (1991)
37. Nardi, B.A., O'Day, V.: Information ecologies : using technology with heart. MIT Press, Cambridge, Mass. (1999)
38. Fredriksson, M.: Online Engineering: on the nature of open computational systems. PhD thesis, Blekinge Institute of Technology (2004)

Part III

Conclusions and Future Work

Results and Future Work

1 Results

In this section we will present the results that has emerged throughout the thesis work in a condensed and combined manner. We will return to the research questions and go through and take a look at how these has been addressed.

1.1 What concepts should be part of a trust framework for online services? (RQ1)

We introduced a trust framework constituting of a number of concepts. We have used the notion of e-service and online service throughout the work of this thesis to specify the whole abstract system and the contexts where a particular delivered service is used. We have specified the following components as part of an online-service [1]:

- *Actors/Stakeholders.* In this thesis we have used both the term *actors* and *stakeholders* to denote an individual, group of individuals, and/or organization(s). We have decided to use *stakeholders* for this purpose exclusively in future work. By identifying stakeholders online-service designers will have a starting point for who they are to design the online service for with respect to trust.
- *Trust concerns* is the basis on which solutions can and should be derived in order to address the problem of trust. Trust concerns are related to at least one actor/stakeholder and is something that concerns the stakeholder with respect to trust. Trust concerns may change over time and the number of concerns that may need to be addressed may vary over the life-cycle of an online service.
- *Trust aspects* are constellations of trust concerns that from the online-service developers' perspective make sense to organize into the same category. This is thus a tool to structure and reason about trust concerns that are part of a similar category. One trust concern can be related to more than one trust aspect. Examples of trust aspects are: security and privacy. The granularity of trust aspects can be different; the application domain can dictate this. In an e-commerce application one aspect could be specified for e.g. transaction security if this is deemed to be an important aspect to specify and address.
- *Trust mechanisms* are deployable solutions to identified trust concerns. They are related to at least one trust concern and least one trust aspect. Examples of a trust mechanism could be encryption. A trust mechanism consist of two stages, one descriptive stage where the trust mechanism is specified and on implemented stage where the mechanism is up and running. A mechanism doesn't need to be deployed in the software itself, it can also be by means of education or hardware design.
- *Trust signs* are signs suggesting that the concerns are addressed. Signs can be inspected by the observer of an online-service. We can't control the perception process, but the trust mechanisms should be implemented with the intention to reflect signs that are associated with the stakeholders' trust concerns.

– The *context* is a holistic view on the online service which denotes a rather abstract combination of a set of actors, e-services, artifacts, location, and, time. The context also includes other components and factors such as contracts, ownerships, responsibilities, legal frameworks, work practices, and, organizational aspects. By using the notion context we want to emphasize that each and every context is unique, e.g. each point in time makes the context unique. On the other hand the highlighting of certain notions enables us to address parts of the uniqueness but still take into consideration the complex and unique nature in each context.

1.2 How are the concepts of a trust framework interconnected and used in a methodology? (RQ2)

Figure 1 gives an overview of how the concepts described in (RQ1) can be interconnected: The following Figure captures the main ingredients of our framework supporting design, implementation and monitoring of trustworthy online-services.

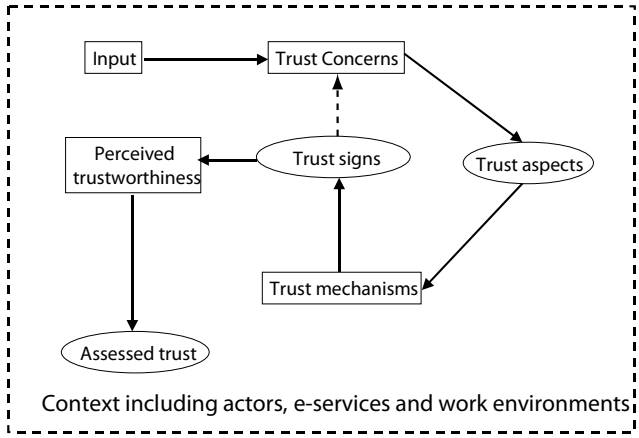


Fig. 1. An illustration of the conceptual relations between our framework components

A trust concern relates to one or more aspects and vice versa. As figure 1 implies there is a cyclic relation between the notions, that is trust concerns are connected to trust aspects, trust aspects are connected to mechanism which in turn are connected to trust signs. Ideally the trust mechanisms are designed in such a way that the trust concerns are addressed in order to support trust assessment for the desired stakeholders. The online service context is the context where stakeholders use, composes and produces the online service. Thus the online service context is a complex notion that can involve a vast number of stakeholders, artifacts, locations and regulations.

1.3 What tools can be designed to support an informed trust design and maintenance of online services? (RQ3)

A trust management cycle and the framework developed serves as tools for the work has been developed and validated:

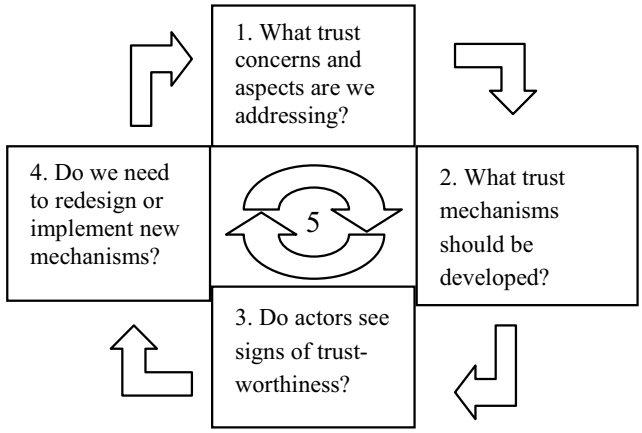


Fig. 2. A trust management cycle used to support design and maintenance of trustworthy online services

The trust management life cycle introduces an approach to deal with the trust framework throughout the life-cycle of an online service. The following steps are part of the trust management cycle:

- *Step one: from trust concerns to trust aspects* – The first step is to identify trust concerns. The dynamic nature of both trust concerns and perception of signs are forcing us to reflect and constantly re-assess our efforts to comply with the online-service stakeholders' trust concerns. The output of this step results in a list of trust concerns and their relationship to the involved stakeholders. The trust concerns should also be connected with one or more trust aspects to provide for means to find general solutions to similar concerns. We summarize step one as: *What trust concerns and aspects are we addressing?*
- *Step two: from aspects to mechanisms* – Trust aspects is just a tool for online service designers to structure the trust concerns into something that opens up the actual activity of finding and building solutions for the identified concerns. This process include efforts to discover, engineer and develop appropriate trust mechanisms. The mechanisms should be connected to one or more trust concerns in order to mediate signs for trust assessment. We conclude this step as *What trust mechanisms should be designed to present signs related to the trust concerns and aspects?*

- *Step three: From mechanisms to signs* – With deployed and implemented trust mechanisms in place the question is if the efforts are perceived by the stakeholder for the cause of trust assessment. Since trust and what the observers perceive is subjective; trust can't be enforced. Here signs come in as the denominator for more or less successful implementation of solutions for trust concerns. If online service stakeholders see and perceive the solution as trustworthy this can be seen as a successful solution to one or more trust concerns for that constellation of stakeholders. The third step of the trust management cycle can be concluded as: *Do stakeholders see signs of trustworthiness that covers the identified concerns?*
- *Step four: Assessment* – The re-assessment phase is the stage where a closer reflection upon the online service context and the trust concerns is done. For instance at one point in time privacy may not be an issue but because of changes in attitudes in society and technical advancements these issues can become more important later in the life cycle of an online service. Thus there is a constant need to re-evaluate and assess trust issues and the online service context in order to keep up with stakeholders' concerns. There is also a need to reflect upon the validity of the deployed trust mechanisms; are they still up to date or do they need fine-tuning? Should a particular mechanism be removed? If trust concerns are still in place we may need to introduce new mechanisms or re-design present ones for that particular concern? In some cases e.g. awareness of legal changes or if a particular encryption chosen is hacked we may need to completely replace a mechanism. This step can be summarized as: *Do we need to redesign or implement new mechanisms or address new concerns?*
- *Step Five: Gateway to the next cycle* – The concerns gathered and findings pointing us towards assumptions that trust concerns aren't properly addressed trigger a new round in the model. This step is a checkpoint where information gathered opens up for a discussion about whether or not to initialize a new round in the trust management cycle.

2 Future work

The work with tools to design trustworthy online services has so far only been briefly addressed. Below some examples are mentioned as suggested directions for further research within trust, trustworthiness, online services, and other possible directions based on the material presented in this thesis.

2.1 Validation and Development of the Framework

One important point is to analyze the appropriateness of the developed framework in more complex services over a longer time span. The example used to validate the framework was tested on a rather limited set of small applications and hasn't been used to reason about how more complex services can be treated. Larger studies where the trust management cycle is used could provide more solid validation. In figure 1 there's a slight but important modification compared to the original version presented in paper 2: *Why Trust is Hard: Challenges in e-Mediated Services* on page 67. The dotted line

between signs and concerns is particularly challenging since this association takes place in the observers mind which we are unable to grasp and analyze as such.

Better approaches for finding concerns are also interesting to explore and input from e.g. the Activity Theory (AT) area would be interesting to further discuss and explore. Much work within the AT-field is currently investigating the role computer-based tools in the context of human activity, cf. Nardi's work in [2], together with Kaptelinin [3], and in work by Bødker [4]. These efforts should be investigated both in order to build systems better and to build the right tools (to use AT-terminology) for certain activities.

2.2 Trust Mechanisms and General Solutions

A trust toolbox containing validated patterns of mechanisms that often seem to have a positive influence on (justified) trustworthiness can be developed by trying to identify successful mechanisms across both time and system evolvement and across domains. Some examples of mechanisms that today are assumed to foster trust in online services such as e-commerce includes certifications and encryption cf. [5, 6]. But what if we scatter a web page with assurances that the site is encrypted or safe to use? Probably an excessive use causes negative reactions or other types of concerns. This is also why we stress a division between trust mechanisms and trust signs since a set of separately perfectly implemented and deployed mechanisms together can give users a negative feeling of trustworthiness.

2.3 Other Uses of our Approach

There are similar non-functional qualities such as trust that may need attention in online service development. Notions such as reliability, flexibility and sustainability can possibly be treated by using an approach similar to that introduced in this thesis. During the later stages of this work we have gained interest in the life-cycle of software intensive systems and the associated process to base solutions on a certain set of needs. We refer back to our perception of development of software intensive systems in the chapter *Introduction* on p.7 where the following four stages where identified: what is needed, what should be built, how should we build it? and finally how do we cope with change? It would be interesting to further elaborate on these four questions in the context of trustworthy online services.

3 Concluding Remarks

The design and maintenance of trustworthy online services is a complex task. It is not easy to tackle issues such as trust and trustworthiness during the design of software intensive system. We hope the provided tools and theories in this thesis will make the process to work toward trustworthy online services easier.

References

1. Rindebäck, C., Gustavsson, R.: Why trust is hard - challenges in e-mediated services. In Falcone, R., Barber, S., Sabater-Mir, J., Munindar, S., eds.: Trusting Agents for Trusting Electronic Societies: Theory and Applications in Hci and E-Commerce. Volume 3577. Springer Berlin / Heidelberg (2005) 180–199
2. Nardi, B.A.: Context and consciousness : activity theory and human-computer interaction. MIT Press, Cambridge, Mass. (1996)
3. Kaptelinin, V., Nardi, B.A.: Acting with Technology: Activity Theory and Interaction Design (Acting with Technology). The MIT Press (2006)
4. Bødker, S.: Through the interface : a human activity approach to user interface design. L. Erlbaum, Hillsdale, N.J. (1990)
5. Egger, F.N.: From Interactions to Transactions: Designing the Trust Experience for Business-to-Consumer Electronic Commerce. PhD thesis, Technische Universiteit Eindhoven (2003)
6. Ang, L., Lee, B.C.: Engendering trust in internet commerce: A qualitative investigation. In: the ANZAM 2000 Conference, Sydney, AU (2000)

Part IV

Appendix

Appendix A: An Online Service for Trusted Delegation of Tasks

This Appendix is based on material from the Alfebiite project[1]. The work presented here is describing a Prolog-based validation system for task allocation and solution to keep track on roles, responsibilities and competences of health care personel. We present the application and some of the underlying Prolog rules. Contributions to this work has been made by Professor Keith Clark, Professor Rune Gustavsson, Christer Rindebäck, and Ashild Steinhovden.

1 Requirements for a new distributed health care system

In Sweden there are two main actors providing health care services, the county council and the municipality. In fact, a third actor, private health care, is becoming a new important factor in health care services. But for simplicity we only take into account the two main actors in our context. In Sweden there is a public health care operated by two main stakeholders; the county council operating what is called primary care, this includes operating hospitals, and local care units. Primary care is responsible for advanced treatment, e.g., when somebody needs acute care due to heart problems or other specialist care including surgery and more advanced treatments. The second stakeholder is the municipality where so-called secondary care is provided. This includes support to the elderly in their daily life with medication and sore treatment as well as assisting parents with questions regarding their children. It also includes examinations and treatments where specialist competence isnt needed. Patients are often treated by actors from both stakeholders. After a surgery has been performed secondary care units may investigate and follow up the recovery progress and inform doctors within the primary care upon progress. Also the surgery would require certain tasks to be done involving daily medication and other kinds of care that is carried out by representatives by another institution.

IT-support has been implemented in order to support seamless distribution of information both within and between involved organizations. This work has mainly been targeted at supporting the distribution of patient information and not on tools supporting distribution of tasks and staff related competence issues. For instance, the casebook of a patient is transferred from a secondary care unit electronically when a patient is hospitalised. After the treatment the procedure has to be repeated and the updated casebook is sent back. The anticipation that more and more patients will be transferred between various stakeholders puts new pressure on the health care actors in order to maintain good quality of care. More and more doctors, nurses and nursing assistants will need to cooperate in order to ensure good quality of care. With todays shortages of trained personnel the competition among actors will cause staff to make multiple career changes

[1] A Logical Framework for Ethical Behaviour Between Infohabitants in the Information Trading Economy of the Universal Information Ecosystem IST199910298. A EU 6th Framework Research Project

and start working for another actor or for a staff service company, which also complies with current laws and regulations. This requires that we better understand who is permitted to do certain tasks based on role related and certifiable skills. This is especially important in environments where staff are constantly changing. The number of tasks that are to be done in order to assure good care is also increasing due to larger number of patients. This, in combination with the increased distribution of health using home-based help care-services, puts extra workload on the involved actors to ensure that all tasks are carried out by people with the right skills. Furthermore, the patient has to remain confident that he will get high quality health services during his lifetime. The issuer of a task has to assess a number of questions such as:

- Has the person being asked the formal competence to carry out the task?
- Have they the practical competence to carry out the task?
- Do I think this person can do this task without negatively affecting the quality of service?

The formal competence to carry out a task is defined by the role of the person being asked. If somebody is a doctor s/he is able to carry out a set of tasks associated with the role of doctor. For instance a doctor has the formal competence to perform a diagnosis and issue prescriptions. Nursing assistants and nurses are not permitted to carry out this kind of task, since they dont have the formal competence to do so. Nurses and assistant nurses have other role related skills that allow them to be allocated tasks requiring such skills.

A practical competence is an on-the-job acquired skill. A particular nursing assistant may have special competence in bandaging sores even though this is outside her formal competences. Her bandaging may have previously always been supervised and checked by a nurse or doctor, but at some point a doctor or a nurse may give the assistant nurse a competence or skill certificate allowing them to do bandaging without supervision. The nursing assistant then has a certified new practical competence allowing her to be assigned bandaging tasks on her own. A certification is a document in which one agent states that another agent has the practical competence to carry out tasks based on a specific skill. By requiring and allowing some skills to be validated, on-the-job, a mechanism for ensuring greater quality of care is in place. The skills for which an agent can issue a certificate depend on the roles of the issuer and the receiver of the skill certificate.

A certification could for instance be offered for the area give insulin injections by Greta with the role doctor to a nursing assistant Emma. The skill certificate is valid for a certain time or until it is withdrawn. This conditional information is also stated in the skill certificate. In order for an offered skill certificate to be valid the receiver must also accept it. The reason for this is that the receiver can have objections whether or not s/he actually has the stated skills. If an agent A allocates a task requiring a skill S to someone with a skill certificate for S, A is still responsible for how well the task is carried out. It is thus not the person with the certified skill who is responsible if something goes wrong. If for some reason the issuer of the skill certificate stops working as a doctor or nurse the skill certificate is withdrawn. An agent who had a skill certificate to give insulin injections by a person no longer working within the issuing institution isnt allowed

to carry out tasks related to that particular skill before receiving a new certification. Skill certificates are solely related to the person for whom they are issued and are non-transferable to other nurses or nursing assistants.

Today information about skill certificates is mostly stored at one location in a binder at relevant care units where the information can be accessed. Due to the increased complexity and number of involved staff members within the healthcare sector there is a need for improved and more flexible record keeping regarding health care skill certificates. This improvement is also needed because health related tasks increasingly take place in the homes of patients using ad hoc home help services Institutions. There is an obvious need to ensure that such ad hoc teams have the necessary skills for the required care. As a matter of fact, the acceptance by all stakeholders, including the patient, of eHealth-to-Home is critically dependant on an accepted sustainable solution of those matters.

2 Elaboration of the role based powers

The above background involves insights gained in discussions with health care staff and on the DICE implementation. However, DICE does not handle skill certificates. Instead the relation between what skills an agent has, and hence the type of tasks they can do, was fixed in simple role based tables. In order to handle the concepts of certification, and to experiment with the idea of electronic certificates, we have replaced the simple table lookup with queries to a Prolog module.

The Prolog module has event calculus rules that can handle the dynamic allocation of new powers and, more generally, is able to keep track of the current state of the system who has been allocated which task, who has been offered and accepted which skill certificates, which responses to requests are outstanding, etc. It therefore handles certificates and the power that they confer on the recipients in a trustworthy way. The table in fig.1 gives an example of some of the role related and certifiable skills of the Swedish health care system. It depicts the relation between different roles and the skills they can have or can certify. Each agent has all the skills of the roles below them. The role levels are in descending order: doctor, nurse, nurse assistant.

3 The DICE2 Application

In this section we will introduce the DICE2 Application including it's features and the underlying technololyg. A scenario will also be presented illustrating how the application can be used.

3.1 The DICE2 Architecture

The intended architecture of the DICE2 application is depicted in Figure 2. The running DICE2 application will be tightly coupled to a Prolog engine running an event calculus representation of the rules regulating the distribution of health care tasks. For instance, when somebody accepts a task a happens fact recording this event is asserted to the Prolog rule base.

Role	Role related skills	Skills that can be certified
Doctor Has power to issue certificates for all skills a nurse or an assistant nurse can perform with a certificate. Has power to announce tasks requiring any role related or certifiable skill of a nurse or assistant nurse Have power to accept tasks requiring any of their role related skills. and any skill for which they can issue certificates	Issue prescriptions Perform a diagnosis Also has all the skills role related and certifiable skills of nurses and assistant nurses	
Nurse Has power to issue certificates for all skills an assistant nurse can perform with a certificate, except insulin injections Has power to accept tasks requiring skills that are included in their role related skills, or the skills for which they can issue certificates, or the skills for which they have a certificate	Give morphine injections Give medication according to prescription Give some medications without a prescription All role related and certifiable skills of assistant nurses	Issue prescriptions of pain killers (requires certification for skill: issue prescriptions of pain killers)
Nursing assistant Has power to accept tasks requiring skills that are included in their role related skills, or the skills for which they have a certificate	Change bandaging on sores Read and report body temperature	Give certain prescribed medications to patients Place catheter on patient Give insulin injections

Fig. 1. A table of role related- and certifiable skills.

Our Prolog program will have rules describing of the roles and the associated role based powers to capture the essential features of the table of fig.1. This is augmented with an event calculus Prolog model, using concepts from the ICSTM social simulator, to capture the dynamic powers, obligations and abilities of the different participants, as certificates and tasks are issued and accepted.

To test the feasibility of this approach, and to experiment with how a Prolog engine could be embedded in the Java DICE2 environment, we have developed a substantial Prolog program[2] and a Java harness that enables us to simulate the event notifications and queries that would be passing between DICE2 and a Prolog engine, Figure 2. For this DICE2 simulation, we have both simplified and elaborated the identified requirements. The simplification is that we have abstracted from the actual skills associated

[2] For a full account on the software developed we refer to the Alfebiite Delivery 8 Documentation

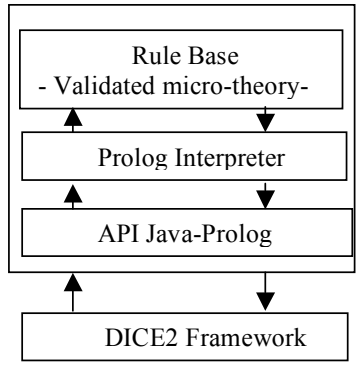


Fig. 2. The DICE2 application

with each role, and we denote them using letters such as a, b, c etc. The elaboration concerns the adding of a new role based power to doctors.

You will notice that in the fig.1 table, nurses cannot issue certificates to assistant nurses for the skill: insulin injections. Such skill certificates can only be issued by doctors. To some extent, the issuing of a skill certificate is a transfer of power. Where the issuer also has the skill they are certifying, which is a constraint satisfied by in our powers and skills fig. 1 table, the issuing of a certificate is an endowing of one of the issuers powers the power to accept tasks requiring the certified skill to the recipient. Following a suggestion of Andrew Jones, that it would be interesting to further investigate the representation of the transfer or endowing of power, we decided to add the concept of the power to issue certificates to certify. An example of this is a power that we could give to doctors to certify others (nurses) to issue certificates (to assistant nurses) for the insulin injection skill. So, in addition to the power to issue skill certificates, our rule set allows doctors to issue certificates to certify others for specific skills. This concept is not in the current Health Care System, but by demonstrating that we can formalize it, and embed the formalization in a distributed Health Care system (DICE2), it could be adopted as a new idea for improving trust and quality of care.

The testbed is developed according to Figure 2. It consists of four parts:

- A rule base where information about powers, permissions and obligations for agents using the system is stored together with the scenario of preceding events
- A Prolog interpreter running the rule base.
- A Prolog-Java Interaction interface which facilitates the interaction between a Java Client and the Prolog engine and rule base.
- A GUI enabling us to simulate the DICE2 application.

3.2 Prolog description of the roles and static role based powers

This part of the Prolog rule set contains facts and rules that enable use to infer the fixed role based powers of the different individuals that are allowed to participate in the dis-

tributed health care, For example, it contains facts such as:

```
password('John S',1234)
password('Jannicke S',7890)
has_role('John S', Doctor')
has_role('Jannicke S', 'Nursing Assistant')
higher('Doctor','Nurse').
```

It also has rules such as:

```
above(Ag1,Ag2):-
      has_role(Ag1,R1),
      has_role(Ag2,R2),
      higher(R1,R2).
```

Allowing us to infer that John S','Doctor' is higher than' Jannicke S'. This is a key property that constrains the issuing of certificates and task requests.

The role related powers are corresponding to the static powers of the earlier table are captured by simple rules such as:

```
role_related_power(Ag,announce_task_for_skills([a,b,c,d,e])):-
     has_role(Ag,'Doctor').


role_related_power(Ag,accept_task_for_skills([b,c,d,e])):-
     has_role(Ag,'Nurse').


role_related_power(Ag,award_certificate_for(Skill)):-
     has_role(Ag,'Doctor'),
     member(Skill,[d,e,f,g]).


role_related_power(Ag,accept_certificate_for(Skill)):-
     has_role(Ag,'Nurse'),
     member(Skill,[f,g]).


role_related_power(Ag,accept_c2c_for(e)):-
     has_role(Ag,'Nurse').
```

The last rule says that every nurse has the power to accept certificates to certify others for skill e (say insulin injections).

3.3 Event Calculus Description of Dynamic Powers, Abilities and Obligations

In a running DICE2 application events happen. For example, doctors and nurses try to assign task by sending request messages using their PDAs, others accept such task request, and execute the tasks. We want to constrain this distribution of work so that it conforms to the institutional rules. We also want to keep track of what has happened by sending event notifications to the Prolog engine that records the events as happens facts. Each time there is a new event, we want to be able to check that it is permitted in accordance with the rules and the scenario of preceding events. More than that, we want to be able, optionally, to constrain the PDAs of the doctors, nurses and assistant nurses so that the action options they are presented with are just those that they are permitted to do at that time. We also want to keep track of each persons current obligations, for example an obligation to perform a task they have accepted. This would be reflected in a to do list on their PDA. For inferring at each moment in time each persons current abilities and obligations we use the event calculus, as in the ICSTM social simulator. The Java interface then repeatedly queries and sends updates to the Prolog engine so that they are synchronised after every event.

Our application specific rules of the event calculus enable us to infer, after any sequence of action events by doctors, nurses and assistant nurses, what the current powers, abilities and obligations of each person are. For us a power is an abstract ability. It is not a power to do a specific action, such as to send a message to some named person, as it is in the ICSTM rules for the contract net. We call a power to perform a specific action an institutional ability for which we use the can modality. If a person can do something, they are have power and permission to do it, and it is a physical action such as sending a message or treating a patient.

Shortly we shall give rules allowing is to infer ability to do a specific action from powers. We first need two rules about powers that an agent has at a particular time:

```
role_related_power(Ag,award_certificate_for(Skill)):-
    holdsAt(pow(Ag,announce_task_for_skills(NeededSkills)),Time):-
            role_related_power(Ag,announce_task_for_skills(AnnounceSkills)),
                forall(member(Skill,NeededSkills),
                member(Skill,AnnounceSkills)).


holdsAt(pow(Ag,accept_task_for_skills(Ag,NeededSkills)),Time):-
        role_related_power(Ag,accept_task_for_skills(StandardSkills)),
            forall(member(Skill,NeededSkills),
              (
               member(Skill,StandardSkills);
               holdsAt(pow(Ag,accept_task_becauseOF_c
               (Skill,C_ID,C_Descr)),Time)
              )
            ).
```

The first rule says that at any time an agent Ag has the power to announce tasks requiring NeededSkills if this list of skills is included in the skills for which they have the role

related power to announce tasks. The power to announce tasks does not change over time. The second rule tells us that an agent Ag has the power to accept a task requiring NeededSkills if every needed skill is one for which they have a fixed role related power to accept tasks, or is a skill for which they have a certificate to accept tasks. This will be the case only if there have been events prior to Time in which they have been offered and have accepted such certificates. So the power to accept tasks varies over time because of skill certificates.

Using the above rules, and the one below, we can infer a specific ability for an Issuer to send a task request message to a Recipient:.

```
holdsAt(can(Issuer,request_task(Issuer,NeededSkills,T_ID,T_Descr,Recipient)),Time):-
        above(Issuer, Recipient),
        holdsAt(pow(Issuer,announce_task_for_skills(NeededSkills)),Time),
        holdsAt(pow(Recipient,accept_task_for_skills(NeededSkills)),Time).
```

Such a task request is an action event that the DICE system will notify the Prolog engine of when Issuer does the action. In effect it says that the issuer can request Recipient to perform a task with id T_ID and description T_Descr requiring skills NeededSkills, only if the Issuer has the power, at that time, to announce tasks requiring all the NeededSkills and the person being asked to do the task has, at that time, the power to accept tasks requiring all of NeededSkills.

In fact, as will become clear when the Java user interface is explained below, the issuer will only be able, using the interface, to send such a request if they it can be inferred that from the rules and the record of preceding action events that they can do it. Using the interface the person, say John S, selects the skills required for the task they want to request from a menu. The skills on the menu are all the ones for which they have the power to announce tasks, determined by a query to the Prolog program. Then, having selected the needed skills, another menu offers John the names of all the people currently empowered to accept tasks needing these skills to whom he is enabled to send such a request. Again this list of people is determined by a query:

```
findall(Ag, holdsAt(can((John S,request_task(John S,NeededSkills,_,_,Ag)),now)
```

sent to the Prolog engine. Here, now is the current clock time. When the permitted action is performed, the Java interface sends the Prolog engine an event notification that results in the Prolog update:

```
assert(happens(request_task(Issuer,,,,,,,,,,Recipient)),now)
```

Here, now is the new current clock time, Issuer is the name of the person sending the task request, Receipient the name of the person to whom it is sent.

The request to do the action now appears on Recipients PDA. They have an obligation to respond, indicated by the options they will have on their display. This obligation is a result of the event rule:

```
initiates(request_task(Issuer,NeededSkills,T_ID,T_Descr,Recipient),
```

```
oblig(Recipient,respond_to_task(ToAg,Answer,NeededSkills,
                T_ID,T_Descr,Issuer)),Time):-
                holdsAt(permission(Issuer,request_task(NeededSkills,
                T_ID,T_Descr, Recepient)),Time).
```

Notice the precondition of this rule says that the issuer has to have permission to issue the request for the obligation to ensue. We have general rules allowing us to infer permission from abilities and obligations:

```
holdsAt(permission(Ag,Action),T) :- holdsAt(can(Ag,Action),T).
holdsAt(permission(Ag,Action),T) :- holdsAt(oblig(Ag,Action),T).
```

We use permission constraints as preconditions of all our initiates and terminates rules because it does not matter whether the action mention in the rule was an obligated action or just an action for which the actor had the rule authorised ability. We can now infer the Recipients obligation to respond using the general event calculus rules:

```
holdsAt(Fluent, Time) :-
        happens(Action, ATime),
        ATime ¡ Time,
        initiates(Action, Fluent, ATime),
        \+ clipped(ATime, Fluent, Time).
clipped(Time1, Fluent, Time2) :-
        happens(Action, Time),
        Time1 ¡ Time,
        Time ¡ Time2,
        terminates(Action, Fluent, Time).
```

The Recipient will be obliged to respond, and will continue to be obliged to respond, unless another even terminates the obligation (the clipped condition).

A respond_to_task action will bring about this termination, as reflected in the termination rule:

```
terminates(respond_to_task(Recipient,_,NeededSkills,T_ID,T_Descr,Issuer),
            oblig(Recipient,respond_to_task(Recipient,_,NeededSkills,
            T_ID,T_Descr,Issuer)), Time):-
            holdsAt(permission(Recipient, respond_to_task(,_,
            NeededSkills,T_ID,T_Descr,Issuer)),Time).
```

A positive response will also initiate an obligation to do the task:

```
initiates(respond_to_task(Ag,'yes',NeededSkills,T_ID,T_Descr,Issuer),
        oblig(Ag,do_the_task(Ag,NeededSkills,T_ID,T_Descr,Issuer)),
        Time):-
        holdsAt(permission(Ag, respond_to_task(Ag,'yes',
        NeededSkills,T_ID,T_Descr,Issuer)),Time).
```

This obligation will be reflected in an entry in the todo list on the responders PDA. The above rules should give the flavour of our representation. We also handle withdrawing of skill certificates and withdrawing of certificates to certify. The former terminates, amongst other things, the power of the person who held the certificate to accept tasks needing that skill and their obligation to respond to any task offers needing the skill to which they have not yet responded. However, if a task has already been accepted on the strength of a certificate that is being withdrawn the obligation to perform the task is not terminated as the person might be on their way to do the task. The withdrawal by a person P of a certificate to certify from a person A results in the termination of As power to offer such certificates and in the termination of As power to withdraw them. The power to withdraw certificates issued by A is transferred to P. The idea is that P should now re-examine each such skill certificate award and withdraw it explicitly if need be.

3.4 The Java GUI

Our intention in connecting the Prolog base to a JAVA environment is to demonstrate proof-of-concepts and validations.One way we do this is to limit the physical capabilities in the application so that the different agents only do things that are in accordance with their powers and obligations at the current time point. To test the feasibility and usability of such a regulated home health care system, we have developed a Java GUI which simulates the distributed task and certificate allocation of an extended DICE system. The interface interacts with a Prolog engine running the previously described event calculus rules sending it event notifications and queries. The GUI has two levels of use. A general administrators level enables us to switch between the views of each worker using the system, Figure 3. The second level allows the system to display the individual view for each user of the system. The individual vies can be displayed, e.g., on the PDA the user is logged on with.

All the events generated by the use of the interface can be inspected from the general view GUI window. For instance, the event generated in fig.4 happened when Petra P signed on (the clock time of this event is given in milliseconds).

We can choose to issue a task from the view of doctor Hilda G or we can choose to issue a task from the view of nurse Petra P. The system support inspections of how powers, permissions and obligations (reflected in our Prolog rule base) effect each persons view of the system. Furthermore, we can inspect current powers, permissions, and can-statements of the application at hand.

When a task request has been issued we can inspect how the requested agents PDA (5)is responding. We will see that the task request will be displayed, and will remain displayed, until a proper response is given.

3.5 A scenario based walk through

Petra P is a nurse at secondary care unit. Her work is concerned with medical treatment of elderly unit providing home help service for persons requiring special needs. The work carried out by home help service staff is highly mobile especially in less densely populated areas on the countryside. The staff members of Petrass department usually

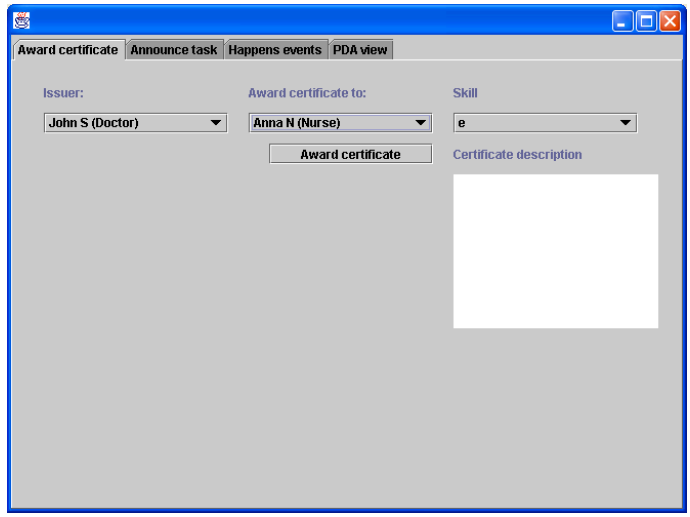


Fig. 3. A screenshot displaying the administrators view that doctor John S is in the position to award a certificate to nurse Anna N with skill e.

drive between the people in need of care according to a daily list of tasks and visits. Some of the people receiving care are visited two or three times each day whilst others occasionally during the week depending on the regularity of the tasks that are to be carried out. The work includes medical tasks as well as taking care of household tasks such as cleaning and taking care of shopping. Due to the amount of less medical advanced tasks involved in many of the tasks the work is usually done by nursing assistants. The coordination of information is largely done via memos and documents stored in binders distributed to the involved staff members. Daily meetings are also held when the shift starts their work for the day in order to coordinate the daily. Petra P plans the work for a number of Nursing Assistants in the care unit located in a less densely populated area of a medium sized municipality in Sweden. She coordinates the work for a number of districts together with another nurse under supervision of two doctors who are consulted regarding medication and diagnoses. Furthermore she prepares medication for all the patients that receive prescribed drugs by putting the required doses in to units containing the exact doses for each and every event for medication. Due to the flue many nursing assistants are home on sick leave so Petra has to make use of the staff substitution pool available within the municipality. Besides one nursing assistant currently available to Petra, one more is assigned to work within Petras group. Because of her coordination role, Petra cannot herself visit all the patients who need assistance with taking their medication. Instead she has to rely on skills possessed by nursing assistants who formally arent allowed to carry out the task of giving patients their medication. However this is a certifiable skill for which Petra can award skill certificates to nursing assistants. The nursing assistant from the staff substitution pool is unknown to Petra so she doesnt know what additional skills she has. Since it is important for Petra to comply

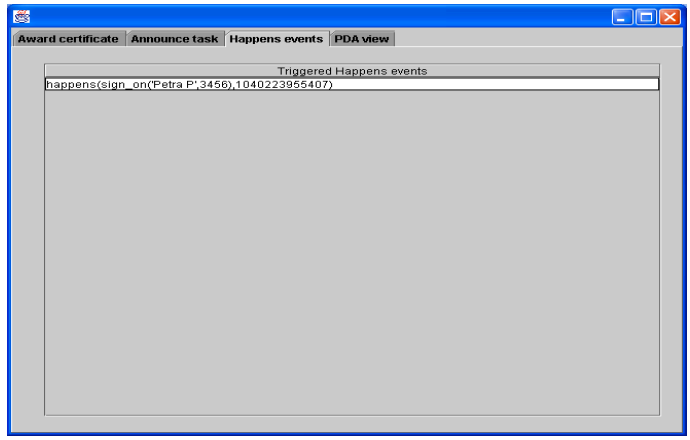


Fig. 4. Logging of events, with timestamps, in the system

with the rules in order to provide trusted care for the care receivers and their relatives, compliance with the rules is an important concern for her. By use of the extended DICE version based on powers, permission and obligations Petra can investigate the skills status of all her the assistant nurses when she is about to issue tasks to them. In our simulation we denote giving medication this skill as c, the corresponding view of the DICE platform can be seen in figure 6.

When Petra is planning to issue a medication task she discovers that none of the new nursing assistants available has a skill certificate for giving medication. She decides to investigate their competence in this matter by observing them giving a number of patients their medication. She decides that the new assistant Jane K has the requisite skill and awards a skill certificate to her (fig. 7)

In order for the skill certificate to be accepted and finally added to her existing skills Jane K must consult her PDA (fig. 8)where the offered skill certificate will be accessible. In our logical representation we say that the agent receiving the skill certificate request is obliged to respond to it in order to comply with the regulations and norms of the organization. When she responds, her response the response event recorded as a happens fact in the Prolog program, including the response of either Accept (Yes) or Decline (No).

Let us suppose that Jane accepts the certificate. As soon as the happens fact has been asserted into the Prolog program the fact that Jane K has this certified skill will be reflected in the PDA of nurse Petra. Returning to the task-issuing Petra P can now see that that nursing assistant Jane K can be asked to do the medication task. The view on the PDA for this is illustrated in 9

Petra P can now issue tasks based on the new skill certificates. Jane K can choose to accept or decline the task using her PDA (fig. 3).

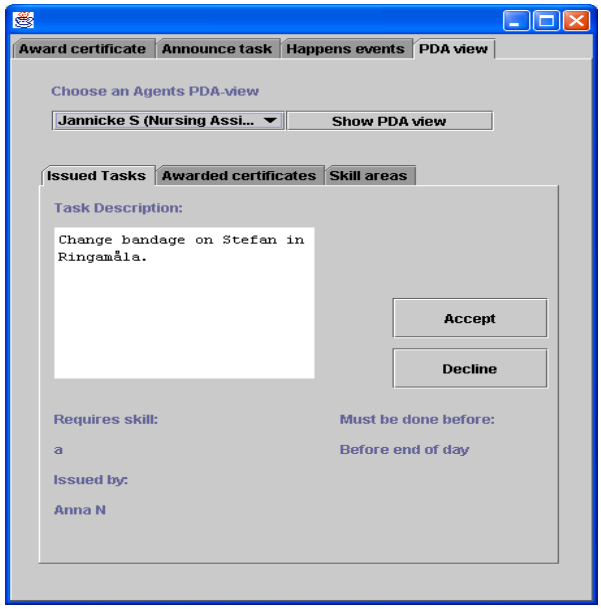


Fig. 5. A picture of Nursing assistant Jannicke Ss PDA showing her that a task has been allocated to her and other relevant information.

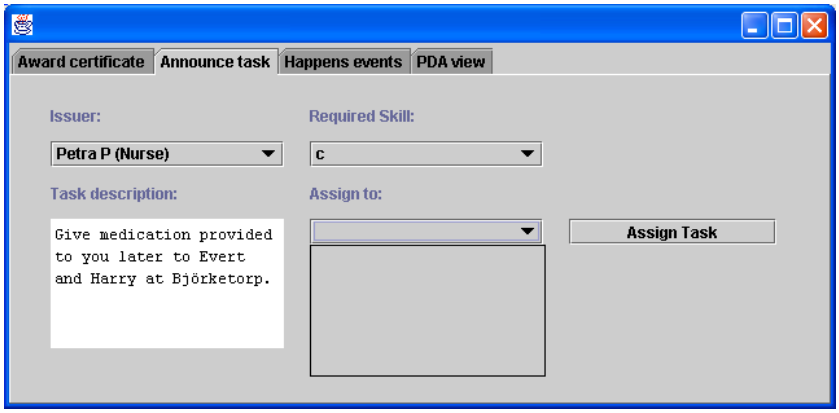


Fig. 6. The nurse Petra P is announcing a medication task belonging to her skill level c.

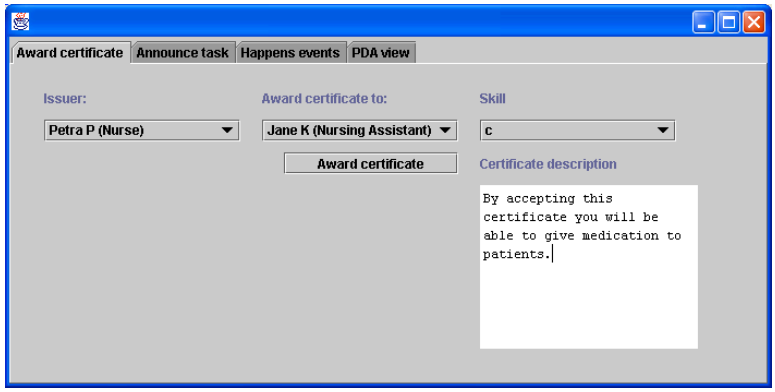


Fig. 7. The nurse Petra P issues a skill certificate to Nursing assistant Jane K.

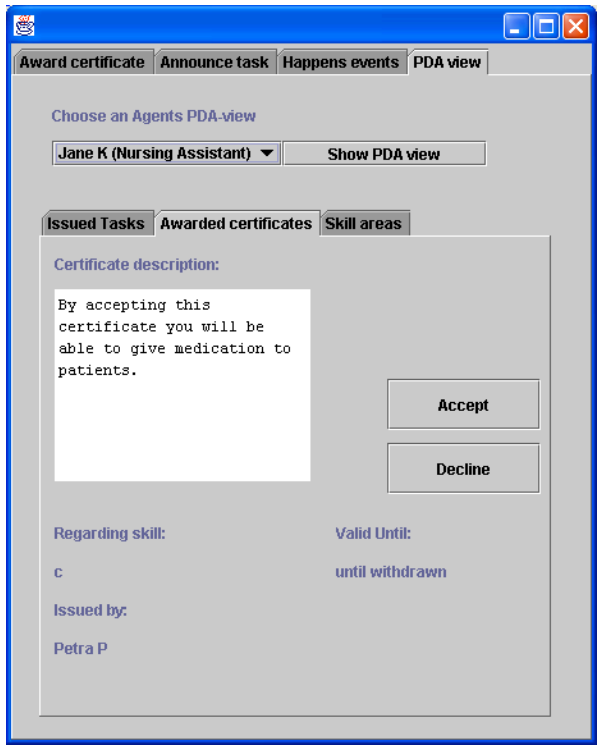


Fig. 8. Viewing the awarded certificate request on Jane Ks PDA.

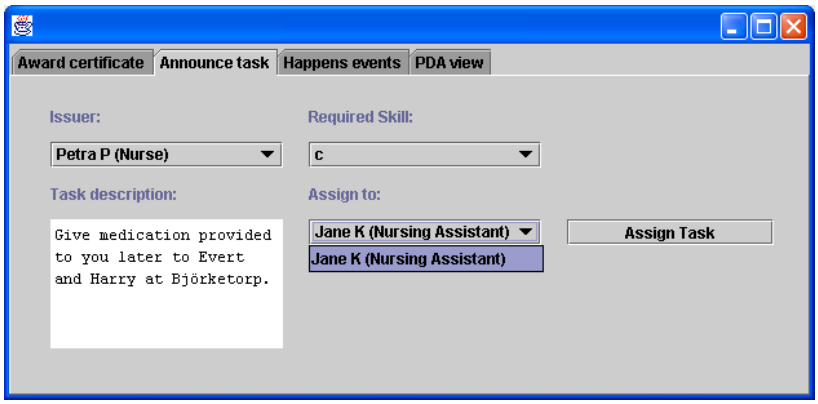


Fig. 9. The nurse Petra P can assess that Assistant nurse Jane K has required skills to be assigned a specific task.

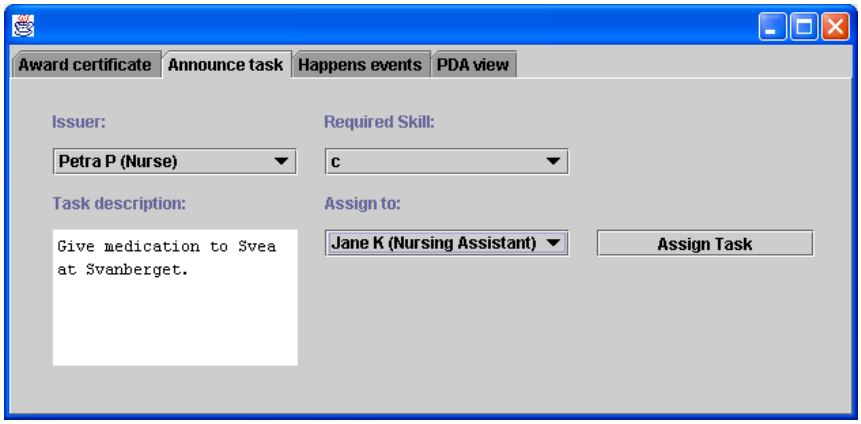


Fig. 10. Controlled announcement of a specific task. Jane K is now *obliged to respond* to the task request, but she is not obliged to accept the task itself.

ABSTRACT

Trust and trustworthiness are two notions that have been discussed extensively in the computer science community, e.g. trust in online banking services. We argue for a broad view on trust, namely trustworthy behavior of online services. We propose solutions enabling online service developers to reason about, and deal with issues of trustworthy online services, from concerns to actual implementations, and assessments. The view on trust in this thesis involves viewpoints on what stakeholders can have trust in, and the need to exhibit and suggest trustworthiness in online services. Trustworthiness and other relevant theories are also discussed. Three main results supporting design and maintenance of trustworthy online services will be introduced.

First, a trust framework in the context of online services is introduced, specifying a number of concepts that enhances and clarifies how trust can be addressed. The framework enables an informed analysis, implementation, and assessment of solutions to trust issues based on identified trust concerns.

Secondly we present how the concepts of the framework can be interconnected. The concepts enables us to reason about stakeholders' trust concerns in relation to deployable solutions called trust mechanisms that are implemented in order to exhibit proper signs suggesting trustworthiness. These signs, we argue, serve as input for stakeholders' trust assessment. The interconnected framework opens up for a discussion on how deployed solutions in an online service correspond to certain stakeholders trust concerns.

Finally a tool for online service designers is presented, the trust management life cycle. This is an approach enabling an informed design practice that emphasizes on a trustworthy design of online services. The use of the cycle is illustrated by the use of a deployed online service.

ISSN 1650-2140
ISBN 978-91-7295-120-4

2007:08