Patterns of Trust in Ubiquitous Environments

Bettina Biel
University of Leipzig
Dept. of Computer Science
Applied Telematics and
e-Business Group
Klostergasse 3
04109, Leipzig, Germany
biel@ebus.informatik.uni-leipzig.de

Thomas Grill
Johannes Kepler University
Linz
Dep. of Telecooperation
Altenbergerstr. 69
4030, Linz, Austria
tom@tk.uni-linz.ac.at

Volker Gruhn
University of Leipzig
Dept. of Computer Science
Applied Telematics and
e-Business Group
Klostergasse 3
04109, Leipzig, Germany
gruhn@ebus.informatik.uni-leipzig.de

ABSTRACT

In ubiquitous environments, users are exposed to public spaces and places where they are supposed to interact and provide also private information. In order to enhance user acceptance of such ubiquitous appliances they have to be designed to consider trust and trustworthiness already in the design phase. We focus on regarding trust in early phases and provide tools for designers by describing trust issues through patterns which are made available through design repositories. Such patterns help designers of ubiquitous applications to create designs quicker based on the availability of already proven solutions they can rely on.

Categories and Subject Descriptors

H.1.2 [**Information Systems**]: Models and Principles – User/Machine Systems

Keywords

HCI, Interaction Design Patterns, Ubiquitous Computing

1. INTRODUCTION

In ubiquitous environments, the designer of appliances faces special challenges. Through disappearing computing approaches and the usage of appliances in the public, new design requirements appear. The idea of computers everywhere, connected via a network seems useful, but it imposes serious social and security issues. End-users are aware that systems collect data which can be used to breach one's privacy. The issue of trust evolved with the emergence of ubiquitous appliances and has become an important issue for designing in ubiquitous environments because of its impact on user acceptance.

Often, investigators rather focus on the technological possibilities of implementing ubiquitous applications than on designing these applications for being used and accepted in a real life environment. The acceptance of ubiquitous applications depends on how these systems are designed for trust. But as an abstract concept, trust is immensely difficult to implement.

Our idea is to develop patterns that help designers of such ubiquitous applications to create designs quicker through the availability of already proven solutions they can rely on. The concept of patterns was introduced by Alexander [2] and adapted for software engineering by Gamma et al. [9] and for interaction design for example by Tidwell [22]. They help to identify, record and catalog successful designs to capture knowledge including trade-offs and design alternatives.

We propose *patterns of trust* representing solutions that integrate trust when designing applications in the context of ubiquitous environments. We define [5] and refine the abstract concept of trust for this domain, and using these insights we derive our specific patterns.

In the following sections, we first define trust and parameters of trust in ubiquitous environments. Further we define patterns of trust. We also present an example of trust patterns, and discuss the evaluation of a pattern library which is used to efficiently find a design solution for integrating trust within a ubiquitous application. The last section of this work-in-progress paper concludes about the current state of our work and gives an outlook of the work that we are currently working on.

2. TRUST IN UBIQUITOUS COMPUTING

A wide range of definitions of trust have been proposed by literature [8, 11, 13, 4]. Chopra and Wallace [5] investigated in the various definitions and found a common denominator that is broad enough to cover all necessary aspects of trust.

> "Trust is the willingness to rely on a specific other, based on confidence that one's trust will lead to positive outcomes." [5]

All definitions are based on the three elements of trust. These elements are identified as *a trustee* to whom trust is directed, *confidence* that trust will sustain and *willingness* that the opponent will act on the confidence.

2.1 The trust interaction model

Riegelsberger et al. [21] define a basic interaction model reflecting the relationship between the participants in a trust model. The participants may be human or machines which does not influence the model itself [18, 17].

MoMM 24–26 November 2008, Linz, Austria
Copyright 2008 ACM 978-1-60558-269-6 ...$5.00.

In order to initiate a trust relationship, the *trustor* and the *trustee* send signals based on the elements of confidence and willingness to express to each other that a basic trust situation is applicable.

If a basic trust relationship is established, the trustor starts with a trusting action that is directed towards the trustee. The trustee may react with either denying trust to the trustor or fulfilling the required action. The trustee's decision thus reflects a confirmation or a decline of the trust relationship based on the initial signals.

2.2 Trust in Ubiquitous Systems

We use the definition of trust cited above. Based on this definition, we apply the trust interaction model from Riegelsberger (see section 2.1) in order to obtain a basic framework for analyzing trust in ubiquitous environments.

In this section, we define parameters relevant for trust in ubiquitous systems. These parameters are based on the definition of Chopra [5]) and categorized according to the elements of confidence and willingness.

2.2.1 Nature of trust

The nature of trust is defined (see [5]) by four dimensions which are individual trust, interpersonal trust, relational trust and societal trust. They have certain characteristics in ubiquitous environments.

Individual trust as a person's propensity to trust, is based on the general acceptance of ubiquitous environments.

Interpersonal trust in ubiquitous environment is based on the trust relationship that needs to be established between a trustor (user) and the trustee which in general is the ubiquitous environment and/or appliance. This is based on the acceptance of the trustee by the trustor and on the "personification" of the trustee within such an environment [17, 18].

Relational trust is characterized by typical practices which evolve over time, influencing one person's attitude and interaction with applications in ubiquitous environments. It is often defined by habits and customs.

Societal trust expresses the overall trust or mistrust of a trustee to appliances located in ubiquitous environments. It includes the phenomenon of *trust in trust* [12], i.e. that a person is more likely to trust if the trustee is trusted by others as well.

Amongst the above named influences on trust, the context in which the trust relation is to be established is important. Trust is specific for a particular situation in a particular context [21].

Analyzing the nature of trust affects the design of applications in ubiquitous environment through an initial level of existing trust. This might be regarded through e.g. a design of an ubiquitous application depending on an existing level of trust within a culture where people might be more (or less) worried about the misuse of their data.

2.2.2 Preconditions for trust actions

Regarding preconditions for executing a trust action, trust itself is a relevant factor to the trustor. The trustor *depends* on the trustee and is willing to accept a *risk* when executing a trust action. Such a risk may be that the person might be betrayed. This can result in a misuse of the information sent together with the trust action and/or the decline of the requested service or functionality by the trustee.

Dependence is present when a person needs to use an application. If these appliances are woven into daily life, dependence becomes strong; and due to the interlocking of the applications, users cannot easily gain an insight to understand how a system works. Therefore, it is especially difficult to establish a trust relationship.

The risk when establishing an initial trust relationship is primarily located at the trustor's side. Such a risk is that provided information like personal data could be misused, e.g. user identification, passwords, location, time and tasks that the trustor wants to execute.

The level of *dependence* and/or *risk* defines the trust signals that are sent out before establishing a trust relationship and expresses the willingness to establish such a relationship.

The credulity or gullibility of a trustor is expressed through the believe that the environment or a system will not harm the person. It is implied by putting initial trust into a trust relationship where no sufficient signals to establish such a relationship might be sent out.

2.2.3 Trustworthiness

> "Trustworthiness is the perceived likelihood that a particular trustee will uphold one's trust." [5]

Defining trustworthiness follows a multidimensional approach and compasses more than one of the dimension simultaneously. When combining the model of Madsen [14] with the definition of Chopra [5], we receive a model that depicts the factors of trustworthiness which are based on the following dimensions of *expectations*, *robustness* and *moral principles*.

Expectations.

Expectations of a user who interacts within an ubiquitous environment imply aspects of the party to interact with as well as aspects concerning the environment. Regarding trust, we target the reliability and understandability of the system with regard to the interpersonal and societal nature of the trust relationship in ubiquitous environments (section 2.2.1).

Reliability of the trustee relies on experiences of the trustor with the environment the person interacts in as well as with experiences with the trustor. An example is that a user would possibly prefer to interact with an already known trustee than with a trustee who is unknown and about whom no source of positive reputation is available. Expectations are also based on past observations, i.e. that a system responds the same way under the same conditions, and on how users understood how a system works, i.e. their mental model of the system's functionality [14].

Understandability of a system implies that user know how to interact within a ubiquitous environment. If users can build a correct mental model of how a system works, their expectations and the system's reactions match [20] and the system becomes more predictable. Predictability is the degree to which the trustee's behavior conforms to the trustor's expectations. Consistency [19] of graphical design, functional style of interaction and evolutionary regarding software product lines [7], natural mapping, visibility, simplicity [20] help users understand how to use an appliance. Hence a system should provide *feedback* to indicate that user actions were noticed, how the system reacts to it, and what the system is doing in general [19]. This transparency helps to make interactions clear and less erroneous in order to avoid that the trustor's faith in the trustee might decrease.

Robustness.

Robustness can be defined as the trustee's skill and stability to fulfill the needs of the trustor. The attribute also comprises accessibility, i.e. the system should be accessible whenever needed. Usage or system errors should not result in a non-usable device or system.

Competence implies that the trustee possesses the skill to fulfill the needs of the trustor. Competence is defined by the attributes *correctness*, which means that the system produces correct outputs, and *availability*, as the system should be accessible whenever needed. This attribute is very important because trust in the automation is mainly based upon the user's perception of its competence [16]. It is very closely related to *credibility*, i.e. to what degree the computer system can be believed. If users believe that an appliance lacks credibility they are likely to refrain from using it, leaving no chance to regain its credibility [23].

Fault Tolerance refers to the stability of the system which has to prevent, recognize and handle usage and system errors of the ubiquitous appliance in a way that the appliance remains usable. Fault tolerance has a strong impact on trust in ubiquitous environment as in these environments a lot of heterogenous applications work together and numerous sources of error's that are not easy to predict are available. Fault tolerance is especially important because even small errors have a disproportionately large effect on user's trust perceptions [16]. That is why systems should be designed for error [20].

Moral principles.

Moral principles or ethics are defined by rules existing within one environment. Such rules might express that no parties act in a way that harms another party. In ubiquitous environments, these ethics are implemented through applications acting according to defined rules.

Positive intentions define the relationship between the trustor and the trustee. Parameters like goodwill, benevolence, loyality and motivations describe such intentions of the trustee [5]). In ubiquitous environments, humans or machines have to trust machines that they are free of malicious code which can harm the user's data or the user regarding especially privacy protection, and fraud. Such intentions are exchanged via signals before establishing a trust relationship.

Privacy is a rule saying that no personal information shall be used by a party without the accordance of the other party to provide the information. Such information is for example identification data like username and password that shall be provided by the user. Data that might be collected by various ubiquitous devices and sensors provides another source of personal information that might be exposed to misuse. An example for such data is location, time, tasks and user profiles.

3. PATTERNS OF TRUST IN UBIQUITOUS ENVIRONMENTS

When designing solutions in the area of ubiquitous computing, we often are confronted with the need for communication or the necessity of exchanging information between two participants. Such participants may be human or a machine and depending on the combination of the communication partners a distinct type of trust is being established. [18]

In order to settle a communication between both, a certain level of trust is necessary. Such a trust level needs to be designed and regarded as a requirement and restriction for designing a particular ubiquitous application.

Our approach is to use patterns that reflect these restrictions and requirements and show a way of regarding trust during the usability engineering process. The patterns should help designers of ubiquitous applications to create designs quicker through the availability of already proven solutions they can rely on.

Trust patterns represent solutions that integrate trust when designing applications in the context of ubiquitous environments. They are closely related to our analysis of the abstract concept of trust in ubiquitous environments in section 2. They are evaluated by experts as well through the user interface designer in order to be able to qualify the patterns as valid solutions. One of the main issues is to evaluate the "*trust factor*" defined through the evaluation of the particular trust pattern (see section 3.1.4) When defining patterns in order to provide re-usability of already defined and used solutions the credibility in a solution is an important factor [23].

3.1 Patterns of trust

Dearden [6] defines characteristics for models in human-computer-interaction design as operationality, expressiveness and re-usability. These characteristics describe the pattern as a model and depict the important requirements for a pattern to be fulfilled.

According to this, *operationality* defines the degree to which the knowledge encapsuled in the pattern is useful for the specific task of designing user interfaces. *Expressiveness* defines the quality of how a pattern describes the particular design problem and how clear the solution is given and applicable for designing interfaces. *Re-usability* is the property of a pattern that defines weather a problem described qualifies to be a pattern useful within the task of designing user interfaces.

In order to be able to describe how patterns are generated and considered as a pattern we split the process of defining a pattern into four subtasks.

3.1.1 Identifying a pattern

Welie and Tidwell [25, 22] already showed how to identify patterns in user interfaces. It is not difficult to find patterns by analyzing the solutions used by designers, yet it is hard to identify those patterns that provide a real benefit to the user.

Thus almost every solution to a problem is a valid candidate for a pattern. In order to define the found solution in the form of a pattern, it needs to describe the solution in the appropriate pattern form so that the pattern can be evaluated, compared and thus more easily found and used. [10]

A pattern needs to have a proven solution. Thus the solution should have been evaluated by testing the trust of a design solution described. An essential aspect of defining a pattern is to categorize the pattern appropriately in order to make it accessible. In our case the categorization needs to reflect our dimensions of trust defined in 3.1.2.

3.1.2 Categorizing the pattern

In order to make a pattern reusable and to be able to identify patterns needed during the design process, a categorization of the patterns is introduced and used within the pattern description.

Regarding the definition of patterns of trust an appropriate categorization to reflect the different dimensions of trust shall be given.

We focus on our dimension defined for trustworthiness (see section 2.2.3).

3.1.3 Describing the pattern

In order to describe patterns, we need a pattern form that holds the structure of the pattern and guarantees that patterns can be described and structured in a proper way.

The pattern form used is based on the pattern forms from Tidwell ([22]) and Welie [24] and defined by Grill [10] who combined both and defined a structure that also covers the area of trust.

In order to properly describe patterns of trust we follow the basic model defined by [21] and propose the following steps to describe the solution of a pattern of trust.

– *Exchanging signals* defines the provision of the necessary signals for defining the initial trust that is necessary for the *trustor* to trust the *trustee.*
– *Trusting Action* describes one or more actions the trustor might do in order to express his trust to the trustee and to establish a trust relationship to the trustee.
– *Fulfillment* defines the outcome and thus the reaction of the trustee to the trusting action. The trustee might thus fulfill the request and reply with the appropriate reaction or he might neglect the trust relationship to the trustor.

In order to be able to fully and clearly describe the pattern we further introduce properties covering the dimensions of trust. In table 1 these properties are described. Further guidelines how to understand and use the properties of a pattern of trust are given.

The parameters of reliability, understandability and feedback belong to the dimension of expectations. Correctness, availability, credibility and fault tolerance explain the dimension of robustness. The parameters of security, safety, and privacy belong to the dimension of moral principles.

3.1.4 Evaluating patterns of trust

When defining patterns in order to include them in a pattern repository (see section 4) it is important to prove the quality of the pattern before introducing the new identified pattern to the pattern library.

The pattern community proposes two approaches of evaluating patterns and categorizing the patterns according to its relevancy.

The usage count defines a counter of how often a pattern is used.

The quality indicator is a measure to evaluate the quality of a pattern according to its content. This measure expresses the relevancy for trust. We identified the following methodologies that provide a quantitative measure for trust and thus for the trust relevancy of the pattern.

Table 1: Parameters of patterns of trust

Category	Description
Reliability	The system responds in the same way under the same conditions.
Understandability	The system is designed in a way that the user understands how to use it, provides help and reacts as earlier expected.
Feedback	Every user action should be indicated to the user and result in a reaction of the system, and the system actions should be transparent to the user.
Correctness	The the system must work properly and provide correct outputs.
Availability	The system must be accessible and available whenever needed.
Credibility	The user can believe in the correctness of the functionality and the advice of the system.
Fault Tolerance	A usage or system error should not result in a non-usable device or system
Security	Technical issues avoiding threat for safety or loss of information through e.g. intrusion.
Safety	Threat for people's health or life.
Privacy	The system should not provide personal information without the accordance of the user.

Riegelsberger et.al. [21] described the mechanics of trust and how the trustor and the trustee relates. Their framework depicts the context a trust pattern is described in and thus builds the basis for defining how to evaluate trust and which dimensions are regarded.

Madsen and Gregor [14] developed a tool based on the work of Moore [15] for being able to measure Human-Computer Trust. They defined scales (reliability, understandability, technical competence, faith, personal attachment) using principal components analyses (PCA) and performed a Cronbach α analysis in order to identify the relevancy of the scale items in relation to human-computer trust. They further discussed how to evaluate human-computer trust according to these scales and to define a measure for human-computer trust. By applying the measure to the context described by Riegelsberger we perceive a quality measure for *trust patterns.*

Tseng [23] elaborated about *credibility* as a measurement for trust and showed how an approach can be evaluated according to the degree of credibility and the gain, loss and regain of credibility (see table 2).

The criteria defined by Tseng builds the basis for integrating a measurement of credibility and thus confidence during the pattern evaluation. According to Tseng a trust relationship that is stable over time only may be established if the trustor as well as the trustee perceives a positive fulfillment of the trust-action. Tseng thus introduces a time-based measure that evolves through perceived trust.

Table 2: Four evaluations of credibility – Tseng

	User perceives product as credible	User perceives product as not credible
Product is credible	Appropriate acceptance	*Incredulity error*
Product is not credible	*Gullibility Error*	Appropriate rejection

Combining the two approaches of counting the usage and applications of trust patterns together with the defined quality of the patterns obtained through statistical analysis and evaluate this usage scenarios over time we gain a level of trust that can be categorized according to figure 1 and may be expressed as a *trust factor*. If only one of the different approaches fails the evaluation of the pattern might result in a pattern that does not evaluate to be introduced or kept in a pattern library of trust. This method is in line with the methodology proposed by Alexander [1] who used a significance measure of patterns that was expressed by zero up to two asterisks indicating the invariance of a pattern.

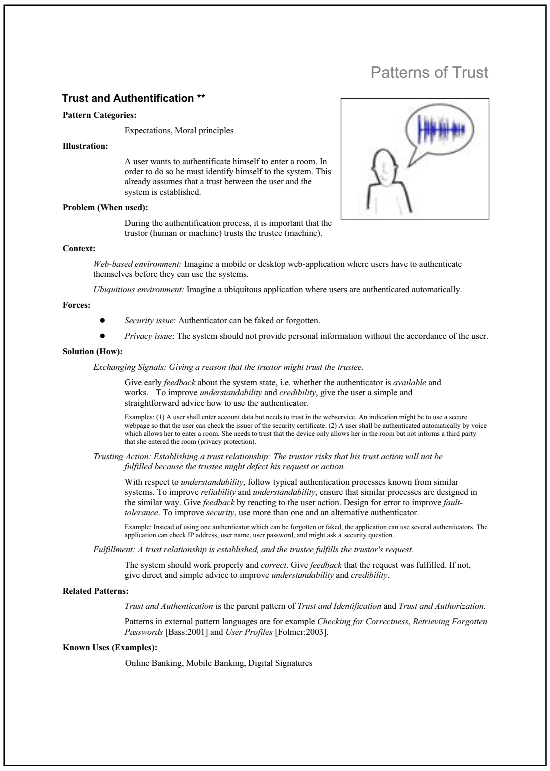
***	a regularly used and evaluated trust pattern
**	a pattern proven to describe a real trust-implementing approach
*	marking a pattern as relevant for trust
No asterisks	a new identified pattern of trust

Figure 1: Pattern categories

3.2 Pattern Example

In this section we give an example of the pattern *Trust and Authentification* which shows how a pattern of trust reflects typical interactions of a trustor with a trustee in an ubiquitous environment as characterized above in section 2. Particularly with the regard of the trust interaction model in section 2.1, the pattern example prescribes which considerations regarding trust are necessary and why. The pattern is categorized as derived from *expectations* and *moral principles*. It describes what an interaction designer needs to consider for *Trust in Authentification* in an ubiquitous environment, where it is important that a trustor (human or machine) can trust the trustee (machine) before, during and after an authentication. We named two forces for the authentification, i.e. security issues because of faked or forgotten authenticators and privacy issues because personal information from a trustor is stored, and the interaction designer cannot ensure that data is not provided to third parties. These two areas indicate that interaction designers and software engineers must work closely together.

The description of the solution comprises three parts, exchanging signals, trusting action, and fulfillment (see section 3.1.3) and includes parameters of trust (see section 3.1.2). In order to describe general applicable patterns, no explicit design proposals are given. Small examples illustrate how a solution might look like. For example, we state, that “for establishing a trust relationship, it is necessary to design an early *feedback* about the system state, i.e. whether the authenticator is *available* and works. To improve *understandability* and *credibility*, it is necessary to give the user a simple and straightforward advice how to use the authenticator.” We also state the two related patterns *Trust and Identification* and *Trust and Authorization.*



Patterns of Trust

Trust and Authentification **

Pattern Categories:

Expectations, Moral principles

Illustration:

A user wants to authentificate himself to enter a room. In order to do so he must identify himself to the system. This already assumes that a trust between the user and the system is established.

Problem (When used):

During the authentification process, it is important that the trustor (human or machine) trusts the trustee (machine).

Context:

Web-based environment: Imagine a mobile or desktop web-application where users have to authenticate themselves before they can use the systems.

Ubiquitious environment: Imagine a ubiquitous application where users are authenticated automatically.

Forces:

- *Security issue*: Authenticator can be faked or forgotten.
- *Privacy issue*: The system should not provide personal information without the accordance of the user.

Solution (How):

Exchanging Signals: Giving a reason that the trustor might trust the trustee.

Give early *feedback* about the system state, i.e. whether the authenticator is *available* and works. To improve *understandability* and *credibility*, give the user a simple and straightforward advice how to use the authenticator.

Examples: (1) A user shall enter account data but needs to trust in the webservice. An indication might be to use a secure webpage so that the user can check the issuer of the security certificate. (2) A user shall be authenticated automatically by voice which allows her to enter a room. She needs to trust that the device only allows her in the room but not informs a third party that she entered the room (privacy protection).

Trusting Action: Establishing a trust relationship: The trustor risks that his trust action will not be fulfilled because the trustee might defect his request or action.

With respect to *understandability*, follow typical authentication processes known from similar systems. To improve *reliability* and *understandability*, ensure that similar processes are designed in the similar way. Give *feedback* by reacting to the user action. Design for error to improve *fault-tolerance*. To improve *security*, use more than one and an alternative authenticator.

Example: Instead of using one authenticator which can be forgotten or faked, the application can use several authenticators. The application can check IP address, user name, user password, and might ask a security question.

Fulfillment: A trust relationship is established, and the trustee fulfills the trustor's request.

The system should work properly and *correct*. Give *feedback* that the request was fulfilled. If not, give direct and simple advice to improve *understandability* and *credibility*.

Related Patterns:

Trust and Authentication is the parent pattern of *Trust and Identification* and *Trust and Authorization*.

Patterns in external pattern languages are for example *Checking for Correctness*, *Retrieving Forgotten Passwords* [Bass:2001] and *User Profiles* [Folmer:2003].

Known Uses (Examples):

Online Banking, Mobile Banking, Digital Signatures

Figure 2: Trust and Authentification pattern

The patterns are closely related to the usability-related software architecture patterns *Checking for Correctness, Retrieving Forgotten Passwords* [3] and *User Profiles* [7].

The evaluation of the patterns was achieved by the usage count and the quality indicator (section 3.1.4). Three examples where “Trust in Authentification” is used comprise online banking, mobile banking, and digital signatures. Regarding its content, the quality of the patterns was evaluated by an expert who marked the pattern “**” because he judged it as proven to describe a real trust-implementing approach which is operational, expressive and re-usable. The pattern’s trust factor will be reviewed after an evaluation project is finished (see section 4). The opinion expressed by the designers and the quality of the outcome will influence the result.

For the use of the patterns we can define the following scenario. Interaction designers have the task of regarding the design factor of trust within an authentification scenario where a person enters a company and faces the necessity to prove his/her identity in order to be able to go to work. The interaction designers first define user characteristics which are related to trust, especially interpersonal and societal trust which are both closely related to a cultural context (see section 2). Resulting from an assessment of dependency and risk, the trustors credulity or good faith can be estimated. Now, the interaction designers may consult a pattern library in order to be able to build a proper design that regards trust as an important factor. To apply the defined patterns of trust when designing ubiquitous applications, several steps are necessary that are based on an interaction design process described in [10]. The patterns in the pattern library are then used to design the interaction. In the process, designers have to discuss their intermediary and finished designs with software engineers who will focus on the implications of the interaction design. In order to support this interdisciplinary co-operation, the patterns of trust already refer to software architecture patterns.

4. CONCLUSION AND FUTURE WORK

In this paper we discussed the factor of trust that plays an important role in designing interactive systems. This applies especially for ubiquitous environments that are stereotypes for applications and appliances that require and/or provide interaction and communication between one or more participants. We further discussed the possibility to use patterns to be able to depict trust as a design factor and to provide proven and reusable examples of how the concept can be implemented within ubiquitous environments. We developed and described such patterns and showed the usefulness and the usage of them for the design of interaction within ubiquitous environments.

Future work comprises the need of developing a more complete range of patterns to depict trust-relevant design approaches and use them within a ubiquitous environment. Further we target to depict the patterns in a pattern map that visualizes the patterns and their categorization which enables designers to use tools to search the pattern library.

5. ACKNOWLEDGMENTS

The Applied Telematics/e-Business Group is endowed by Deutsche Telekom AG.

6. REFERENCES

[1] C. Alexander. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series).* Oxford University Press, 1977.

[2] C. Alexander. *The Timeless Way of Building.* Oxford University Press, 1979.

[3] L. Bass, B. E. John, and J. Kates. Achieving usability through software architecture. Technical report, Software Engineering Institute (SEI), Carnegie Mellon University, 2001.

[4] E. Chang, F. Hussain, and T. Dillon. *Trust and Reputation for Service-Oriented Environments: Technologies For Building Business Intelligence And Consumer Confidence.* John Wiley & Sons, 2005.

[5] K. Chopra and W. A. Wallace. Trust in electronic environments. In *HICSS '03: Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9*, page 331.1, Washington, DC, USA, 2003. IEEE Computer Society.

[6] A. M. Dearden and M. D. Harrison. Abstract models for hci;. *International Journal of Human-Computer Studies*, 46(1):151–177, 1997.

[7] E. Folmer, J. v. Gurp, and J. Bosch. A framework for capturing the relationship between usability and software architecture, 2003.

[8] D. Gambetta. Can we trust trust. In *Trust: Making and Breaking Cooperative Relations*, pages 213–237. Basil Blackwell, 1988.

[9] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns Elements of Reusable Object-orientated Software.* Addison -Wesley, 1995.

[10] T. Grill and M. Blauhut. Design patterns applied in a uid process for safety critical environments (sces). In *Proc. of the 4th Usability Symposium USAB 2008, submitted for review to USAB 2008.* Springer-Verlag, 2008.

[11] F. K. Hussain and E. Chang. An overview of the interpretations of trust and reputation. *Advanced International Conference on Telecommunications*, 0:30, 2007.

[12] J. D. Lewis and A. Weigert. Trust as a social reality. *Social Forces*, 63(3):967–985, 1985.

[13] N. Luhmann. Trust: Making and breaking cooperative relations, 2000.

[14] M. Madsen and S. Gregor. Measuring human-computer trust. In G. Gable and M. Viatle, editors, *Proceedings of the 11th Australasian Conference on Information Systems*, page p. 53ff, 2000.

[15] G. C. Moore and I. Benbasat. Development of an instrument to measure the perceptions of adopting an information technology innovation. *Information Systems Research*, 2(3):192–222, Sept. 1991.

[16] B. M. Muir and N. Moray. Trust in automation. part ii. experimental studies of trust and human intervention in a process control simulation. *Ergonomics*, 39(3):429 – 460, 1996.

[17] C. Nass and J. Steuer. Voices, boxes, and sources of messages computers and social actors. *Human Communication Research*, 19(4):504–527, 1993.

[18] C. Nass, J. Steuer, and E. R. Tauber. Computers are social actors. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 72–78, New York, NY, USA, 1994. ACM.

[19] J. Nielsen. *Usability Engineering.* Academic Press, 1993.

[20] D. A. Norman. *The Design of Everyday Things.* B&T, 1988.

[21] J. Riegelsberger, M. A. Sasse, and J. D. McCarthy. The mechanics of trust: a framework for research and design. *Int. J. Hum.-Comput. Stud.*, 62(3):381–422, 2005.

[22] J. Tidwell. *Designing Interfaces.* O'Reilly, first edition edition, November 2005.

[23] S. Tseng and B. J. Fogg. Credibility and computing technology. *Commun. ACM*, 42(5):39–44, 1999.

[24] M. van Welie and G. C. van der Veer. Pattern languages in interaction design. In *INTERACT*, 2003.

[25] M. V. Welie and H. Trætteberg. Interaction patterns in user interfaces. In *Proc. Seventh Pattern Languages of Programs Conference: PLoP 2000*, pages 13–16, 2000.