

Please quote as: Hoffmann, A.; Söllner, M.; Hoffmann, H. & Leimeister, J. M. (2012):  
Towards Trust-Based Software Requirement Patterns. In: Second International  
Workshop on Requirements Patterns (RePa' 12), Chicago, Illinois, USA.

# Towards Trust-Based Software Requirement Patterns

Axel Hoffmann, Matthias Söllner, Holger Hoffmann, and Jan Marco Leimeister

*Research Center for Information System Design*

*Kassel University, Germany*

*{axel.hoffmann|soellner|holger.hoffmann|leimeister}@uni-kassel.de*

**Abstract**—Users adopt trust to reduce social complexity that can be caused by the lack of knowledge about the inner working of an information system. Our aim is to translate results from trust research about the transformation of user trust in new technologies into software requirement patterns. Therefore, we collect antecedents that build trust, and develop requirement patterns that demand functionality to support these antecedents. This paper presents software requirement patterns consisting of the name, the goal, forces and the pre-defined requirement template that can be used to specify trust based requirements.

**Keywords**—software requirement pattern; trust

## I. INTRODUCTION

Nowadays information systems are increasingly automated and opaque. In general, the users have no ability to know what happens exactly inside a system. This lack of effective regulation of the system generates complexity. People try to reduce social complexity by the adoption of trust [1]. This is caused by the inherent need of people to understand others and their surroundings. By trusting, people reduce their perceived social complexity by a belief allowing the people to act in uncertain environments, albeit this belief may be irrational and may lead to vulnerable behavior [1]. The same argument also holds with new technologies. Users' trust is a key factor for the adoption of new technologies and systems [2].

Unfortunately, trust is a fuzzy concept and there are only few guidelines that help requirements analysts to specify trustworthy systems. Trust usually cannot be enhanced by supplementing individual software components or modules to a system, as trust-based requirements affect the whole software. To speak from one's own experience, early consideration of systematic trust enhancement does not take place in most current development projects.

On the other hand, trust research has produced a vast amount of insights on trust and the formation of trust [3]. This overwhelming amount of insights requires a deep and broad understanding of concepts of trust to deduct requirements for system functionality that enhances the users' trust in the system.

The aim of this paper is to present requirement patterns consisting of the name, the goal, forces and the pre-defined requirement template that can be used in system development projects to formulate trust-based requirements. Therefore, two research questions are answered. First, what

makes a system more trustworthy, and second, how do software requirement patterns to specify trustworthy systems look like.

The remainder of the paper is organized as follows. First, we provide an overview of the related work in trust theory. After a description of the research design in section 3 we present software requirement patterns to enhance user trust in section 4. This is followed by the discussion of our results and conclusion.

## II. TRUST

The understanding of the trust concept itself, the formation of trust and insights on how to foster trust vary across disciplines. In computer science, most of the research on trust focuses on aspects like security, authentication and access control [4]. Other disciplines focus on developing an understanding of trust in human-computer relationships. The focus of trust research in these disciplines lies on understanding trust in social situations. One of the major factors is to work with the users' perception of a system in order to understand why they trust or distrust a system and what can be done to foster their trust. Due to our focus on the latter one we define trust as “the willingness of a party [trustor] to be vulnerable to the actions of another party [trustee] based on the expectation that the other will perform a particular action important to the trustor, irrespective of the ability to monitor or control that other party” [5, P. 712].

Information systems can take two roles in trust relationships. First, they can be used to mediate trust-relationships between humans. Second, they can be used as tools that provide recommendations [6] or control processes [7]. We focus on the latter case where information systems take the role of a trustee in a trust relationship.

Research in trust engineering suggests that trust itself can hardly be addressed [3], and that it is better to address the different antecedents that increase trust. Trust antecedent are factors or elements that build trust. They are often referred interchangeably as antecedents, dimensions, determinants, basis or principles of trust. The antecedents express what is *perceived* by the user. Therefore, it is essential to influence the user perception. Consequently, a technical improvement of a system, e.g., using a better encryption algorithm, will have absolutely no effect if this improvement is not communicated to the users in a way allowing them to understand the benefits for themselves.

### III. RESEARCH DESIGN

The unit of analysis of the first research question is the entirety of trust research in information systems. Many studies have investigated what constitutes trust in a system. We conducted a literature review of the formations of trust. The trust we are interested in is referred as trust in automation, trust in technology, trust in the device, or system trust, where the latter is used for organizational systems and IT systems, and we are only interested in the second kind.

Due to the huge number of contributions on trust and many different proposed antecedents, we build on the results of previous meta-studies collecting trust antecedents. Therefore, we did a systematic literature review in the following electronic libraries: AISEL, ACM, IEEE Xplore, EBSCO BSP, Emerald (Journals), Springerlink, Wiley Interscience, JSTor, PsycAericles, PsycINFO, and Sage. We searched for antecedents, dimensions, determinants, basis, and principles in combination with trust and read the title and abstract to select only the articles elaborating trust in automation. We could identify ten meta-studies [7; 8; 9; 10; 11; 12; 13; 14; 15; 16] including overall 117 publications and reporting a total of 146 antecedents of trust. All these antecedents and the definitions from literature were collected within one list. Due to the use of meta-studies the antecedents were presented in a way we could extract them without interpretive work. Partly, we needed to consult the referred literature in the meta-studies to get a definition for the antecedents. Antecedents with the same name were merged. We did not check if the same antecedents were mentioned with different names in different studies.

We formulated the initial requirements patterns during a period of one year while specifying four automated systems (DinnerNow, MeetU, SupportU, and MyGroup; partly described in [3; 17]) within a larger research project. The first system was DinnerNow. We used the application scenarios to understand the interaction of the user with the system. We reviewed the definitions of the antecedents in the list and checked if it is possible to address this antecedent within system requirement specifications. We formulated software requirements and added them to the system specification.

Further, we used all four trust-based requirements for DinnerNow to follow the opportunistic approach for building requirement patterns [18]. We formulated the pattern goal that expresses what the pattern should achieve. In general the pattern goal is to raise the associated trust antecedents. Therefore, the pattern goal needs to deal with the perception of the user. The goal has the role of the problem part of a pattern. It has an important role since it will help the requirements analyst to decide whether the pattern is applicable to the system [19].

The pattern template is the core of the solution, stating that the software has to achieve the goal of the software requirement pattern. It does not have to indicate how this goal can be achieved [19]. To reach one goal, different templates are possible. The template needs to be adapted to

the system when added to the requirements specification. We formulated the template as simple sentences with the structure suggested in [20]. Finally, we used the antecedents as name for the requirement pattern to summarize the core of the templates in only view words.

For the other three systems we first reviewed the requirement pattern we already formulated. If the goal of the pattern were appropriate for the system we used the template to formulate software requirements. Afterwards, we also reviewed the definitions of the antecedents in the list and checked if it is useful to address other antecedents. If so, we formulated a requirement for the specification and a new requirement pattern. After the specification of the four systems we had collected nine requirements pattern.

With the initial trust-based patterns we conducted an interactive pattern workshop to improve the understandability and helpfulness of the patterns. The participants of the workshop were two requirements analysts (two and five years of experience), two trust expert (three and nine years of experience), and three software engineers (one, two and three years of experience). All of them have a university degree, two hold a doctorate. Only one requirements analyst had known software requirement patterns before.

The procedure of the workshop started with an introduction about the workshop goals, the role of patterns within the requirements engineering process and the specific role of user trust within the development of automated systems. The workshop was conducted using the collaboration software tool ThinkTank by GroupSystems which allows giving feedback anonymously. The software requirement patterns were presented within the tool to the participants.

In the first step of the workshop the participants should read all patterns (including all parts) and judge for each pattern if the pattern is easy to understand, if it is helpful for system development and if they overall would agree with the presented pattern or if there are concerns which need to be discussed in the group. The participants were asked to leave comments or raise issues about the software requirements within the tool. In the next step the participants were asked to read the issues and suggest solutions (as comments in the tool) improving the understandability and helpfulness of the software requirement patterns. In a discussion, all participants checked the issues and solutions and rephrased the parts of the patterns if it was commonly agreed to be useful. Four patterns were discarded after the discussion.

The resulting requirements patterns were afterwards adapted to the RePa requirements pattern template [21]. Therefore we added the RE activity, the pattern type and the stakeholder. Further we described the forces that need to be balanced by the requirements.

### IV. RESULTS

In this section we present the requirement pattern according to the RePa Requirements Pattern Template.

TABLE I. REQUIREMENT PATTERN: EXPLICATION OF INTENTION

<b>Explication of Intentions</b>		
<i>RE Activity: Elicitation, Specification</i>	<i>Pattern Type: Product</i>	<i>Stakeholders: Users</i>
Goal (Problem)	Satisfy the user need of having a system that provides insights about its inner working.	
Forces	Be honest but be aware that the user will not accept every intention. Offer the information in a way, that does not disturb the user during usage, but place it in a way that the users can access it if they want to.	
Template (Solution)	The system shall explicitly display or say that it will act in a particular way.	
<i>Application and Examples</i>		<i>Known Uses</i>
The system shall display information for what the requested personal data is used for.		MeetU [17]

The first requirement pattern is about explications of intentions (Tab. I), meaning the system explicitly displays or says that it will act in a particular way [22; 23; 24]. The system needs to give Information about its inner working in a way that is accessible and available to the average user. These can be information about the purpose of functionality, required data and used algorithms.

TABLE II. REQUIREMENT PATTERN: UNDERSTANDABILITY

<b>Understandability</b>		
<i>RE Activity: Elicitation, Specification</i>	<i>Pattern Type: Product</i>	<i>Stakeholders: Users</i>
Goal (Problem)	Satisfy the user need of having a system that explains what it is doing.	
Forces	There is a tradeoff between showing the users what the application is doing (in the background) and to worrying the them. Hiding information can destroy the trust in the application, if the user discovers hid activities.-	
Template (Solution)	The system shall provide information about current activities.	
<i>Application and Examples</i>		<i>Known Uses</i>
The system shall signal if it is sharing of the current position of the user.		MeetU [17]

Two related trust antecedents are understandability and predictability. Understandability means that the human supervisor or observer (user) can form a mental model of the system [23; 25]. Therefore, feedback to user actions and confirmation of inputs can be used (Tab. II). Due to the mental model the user can predict future system behavior.

TABLE III. REQUIREMENT PATTERN: TRANSPARENCY

<b>Transparency</b>		
<i>RE Activity: Elicitation, Specification</i>	<i>Pattern Type: Product</i>	<i>Stakeholders: Users</i>
Goal (Problem)	Satisfy the user need of easily discern the system.	
Forces	The user wants to know how outcomes, such as recommendations, are generated. There is a tradeoff between securing the mechanisms against competitors and the curiousness of the users.-	
Template (Solution)	The system shall provide information how output was created.	
<i>Application and Examples</i>		<i>Known Uses</i>
The system shall provide the used data for computing the recommendation.		DinnerNow [3]

The mental model can be formed if a system is transparent. Transparency refers to the extent to which the system is clear or easily discerned [24]. Therefore, the system should allow the user to access additional information explaining used the algorithms and data for automated actions (Tab. III).

TABLE IV. REQUIREMENT PATTERN: INFORMATION ACCURACY

<b>Information Accuracy</b>		
<i>RE Activity: Elicitation, Specification</i>	<i>Pattern Type: Product</i>	<i>Stakeholders: Users</i>
Goal (Problem)	The system shall allow the user to select which data sources is used by the system.	
Forces	Users want to have the feeling to influence the outcome of an application. However, sometimes the users do not know the best data sources.	
Template (Solution)	The system shall provide possibilities for the user to select data that is applied by the system.	
<i>Application and Examples</i>		<i>Known Uses</i>
The system shall provide possibilities for the user to select data of friends for the recommendation.		DinnerNow [3]

Information accuracy refers to fact that users will more likely to trust a system that uses accurate, current, and complete data or information [9]. Since the perception regarding sources that are accurate, current and complete may differ across different user, the user should have the possibility to select which sources should be used (Tab IV).

TABLE V. REQUIREMENT PATTERN: PERSONALIZATION

<b>Personalization</b>		
<i>RE Activity: Elicitation, Specification</i>	<i>Pattern Type: Product</i>	<i>Stakeholders: Users</i>
Goal (Problem)	Satisfy the user need of having a system that they can be adapted to personal needs.	
Forces	Tradeoff between automated capturing and setting options, and between additional setting options, and ease of use of the system.	
Template (Solution)	The system shall provide setting options for system functionalities.	
<i>Application and Examples</i>		<i>Known Uses</i>
The system shall allow the user to explicitly rely on ratings of friends for a recommendation.		DinnerNow [3]

Personalization is defined as a user’s perception of the extent to which the system understands and represents his or her personal needs [6]. This means, that users’ trust increases if they have setting options to adapt the system according to their needs. At this level it cannot be generalized which settings are useful, but the requirement suggest that more setting options have a positive influence on trust (Tab V). Of course, other factors, such as the ease of use of the system, should not be influenced negatively.

In the expert workshop four initial requirement patterns were dropped as result of the group discussion. These patterns addressed predictability, familiarity, level of automation, and designer reputation.

## V. DISCUSSION

The requirement patterns were developed while eliciting requirements for automated systems. That is the context we expect the patterns work best. It should be possible to adapt the pattern to other information systems that provide a graphical user interface. The pattern can help requirements analysts to address trust on a basic level. Other approaches like trust engineering [3] can help them to achieve more detailed and maybe more suitable trust requirements.

In the expert workshop we could observe different opinions about the requirement patterns. While the trust experts emphasized the addresses antecedents and requirements as important, some of the software engineers denoted the pattern as trivial. On closer inspection this is no contradiction. User trust can be increased by fulfilling trivial requirements, but they should not be missed in the development project.

The trust-based requirement patterns were found having overlaps with other system characteristics, especially usability. This is in line with research showing that perceived ease of use is also an antecedent of trust [2]. Therefore, we can assume that every effort to enhance usability will also enhance user trust in the system. Nevertheless, the used antecedents were claimed in literature to particularly increase trust.

## VI. CONCLUSION

Results from trust engineering show that trust can be enhanced systematically during the system development process [3]. With the help of the presented software requirement pattern we want to give requirements analysts who want to specify information systems an easy-to-use approach for considering user trust. According to IS theory on technology acceptance, increased user trust enhances the chances that a specific system will be adopted by its intended users [2]. Thus using the presented patterns will help requirements analysts to specify requirements for an information system that enhance the chance of the system of being adopted by its intended users. To identify patterns, we examined different trust antecedents and searched for suitable requirements to address these antecedents. We found technical requirements which were important in different systems. From these technical requirements we formulated the software requirement patterns.

In future research more antecedents shall be addressed. To enhance usability of the requirement pattern in practice we plan to integrate the requirement patterns within an existing requirement pattern catalog. Further, we want to apply known parameterization to allow more detailed choices by each analyst using the pattern and make it easier to adapt the pattern for different kinds of information systems.

## REFERENCES

[1] Luhmann, N., "Trust and power," Chichester, UK: Wiley 1979.

- [2] Gefen, D., Karahanna, E., and Straub, D.W., "Trust and TAM in Online Shopping: An Integrated Model." *MIS Quarterly*, vol. 27(1), 2003, pp. 51-90.
- [3] Söllner, M., Hoffmann, A., Hoffmann, H., and Leimeister, J.M. 2012. How to use behavioral research insights on trust for HCI system design. Proceedings of the 2012 ACM annual conference extended abstracts on Human Factors in Computing Systems Extended Abstracts 1703-1708. Austin, Texas, USA: ACM.
- [4] Artz, D. and Gil, Y., "A survey of trust in computer science and the Semantic Web." *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 5(2), 2007, pp. 58-71.
- [5] Mayer, R.C., Davis, J.H., and Schoorman, F.D., "An Integrative Model of Organizational Trust." *Academy of Management Review*, vol. 20(3), 1995, pp. 709-734.
- [6] Komiak, S.Y.X. and Benbasat, I., "The Effects of Personalization and Familiarity on Trust and Adoption of Recommendation Agents." *MIS Quarterly*, vol. 30(4), 2006, pp. 941-960.
- [7] Lee, J.D. and See, K.A., "Trust in Automation: Designing for Appropriate Reliance." *Human Factors*, vol. 46(1), 2004, pp. 50-80.
- [8] Beatty, P., Reay, I., Dick, S., and Miller, J., "Consumer trust in e-commerce web sites: A meta-study." *ACM Comput. Surv.*, vol. 43(3), 2011, pp. 1-46, doi:10.1145/1922649.1922651.
- [9] Beldad, A., de Jong, M., and Steehouder, M., "How shall I trust the faceless and the intangible? A literature review on the antecedents of online trust." *Computers in Human Behavior*, vol. 26(5), 2010, pp. 857-869, doi:10.1016/j.chb.2010.03.013.
- [10] Hancock, P.A., Billings, D.R., Schaefer, K.E., Chen, J.Y.C., de Visser, E.J.et.al., "A Meta-Analysis of Factors Affecting Trust in Human-Robot Interaction." *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 53(5), 2011, pp. 517-527, doi:10.1177/0018720811417254.
- [11] He, J., "Understanding the Sources and Impacts of Trust in E-Commerce: A Meta-Analysis," *Proc. AMCIS 2011 Proceedings*, 2011, pp. Paper 142.
- [12] Holsapple, C. and Sasidharan, S., "The dynamics of trust in B2C e-commerce: a research model and agenda." *Information Systems and E-Business Management*, vol. 3(4), 2005, pp. 377-403, doi:10.1007/s10257-005-0022-5.
- [13] Kumaraguru, P., Acquisti, A., and Cranor, L.F. 2006. Trust modelling for online transactions: a phishing scenario. Proceedings of the 2006 International Conference on Privacy, Security and Trust: Bridge the Gap Between PST Technologies and Business Services 1-9. Markham, Ontario, Canada: ACM.
- [14] Li, F., Pieńkowski, D., van Moorsel, A., and Smith, C., "A Holistic Framework for Trust in Online Transactions." *International Journal of Management Reviews*, vol. 14(1), 2012, pp. 85-103, doi:10.1111/j.1468-2370.2011.00311.x.
- [15] Papadopoulou, P., Kanellis, P., and Martakos, D., "Investigating trust in e-commerce: a literature review and a model for its formation in customer relationships," *Proc. AMCIS 2001 Proceedings*. Paper 155, 2001, pp. 791-798.
- [16] Shankar, V., Urban, G.L., and Sultan, F., "Online trust: a stakeholder perspective, concepts, implications, and future directions." *The Journal of Strategic Information Systems*, vol. 11(3-4), 2002, pp. 325-344.
- [17] Comes, D., Evers, C., Geihs, K., Hoffmann, A., Kniewel, R.et.al., "Designing Socio-technical Applications for Ubiquitous Computing - Results from a Multidisciplinary Case Study," *Proc. Distributed Applications and Interoperable*

- Systems (DAIS 2012), Springer, 2012, pp. 194-201, doi:10.1007/978-3-642-30823-9\_17.
- [18] Withall, S., "Software Requirement Patterns," Redmont, Washington: Microsoft Press 2008.
- [19] Renault, S., Mendez-Bonilla, O., Franch, X., and Quer, C., "A Pattern-based Method for building Requirements Documents in Call-for-tender Processes." *International Journal of Computer Science and Applications*, vol. 6(5), 2009, pp. 175 - 202.
- [20] Hooks, I., "Writing Good Requirements (A Requirements Working Group Information Report)," *Proc. Third International Symposium of the NCOSE - Volume 2*, 1993.
- [21] Chung, L., Paech, B., Zhao, L., Liu, L., and Supakkul, S. 2012. RePa Requirements Pattern Template v1.0.1. Second International Workshop on Requirement Pattern (RePa' 12). Chicago, USA.
- [22] Sheridan, T., "Trustworthiness of command and control systems," *Proc.*, 1988, pp. 427-431.
- [23] Madsen, M. and Gregor, S., "Measuring human-computer trust," *Proc. 11th Australasian Conference on Information Systems*, 2000.
- [24] Adams, B.D., Bruyn, L.E., Houde, S., Angelopoulos, P., Iwasa-Madge, K.et.al., "Trust in automated systems," Toronto: Ministry of National Defence 2003.
- [25] Sheridan, T.B. and Parasuraman, R., "Human-Automation Interaction." *Reviews of Human Factors and Ergonomics*, vol. 1(1), 2005, pp. 89-129, doi:10.1518/155723405783703082.