Designing for Trust

Michalis Pavlidis

University of East London
4-6 University Way, London, UK, E16 2RD
m.pavlidis@ieee.org

Abstract. Information systems exist in every aspect of our life. Therefore we need to make sure that users trust these systems in order to continue to use them. However, developing a trustworthy system is a challenging task for the right operation of the system as well as system's acceptance from the users' perspective. Ensuring well-placed trust always depends on the trustworthiness of the trustee, i.e. the system. Therefore, properties that belong to the trustworthiness highly influence to the development of trust. Nevertheless, trust has to be considered from a holistic perspective, since it covers a wide range of issues that belong to different disciplines. In this paper, we present the relevant properties of trust and trustworthiness from a holistic perspective and we define trustworthiness as the competence and ability of a system.

Keywords: Trustworthy information systems, systems development, trust, trustworthiness.

1 Problem Statement

Information has a large impact and plays an important role in different areas of human life. Trust of organisations and individuals in information systems is therefore becoming a central issue for the effective usage of such systems and the stored information.

Developing trustworthy information systems is important not only for the right operation of these systems, but also for their social acceptance and the advancement of information networking. So, we envisage developing a methodology to guide the developers through the process of trustworthy information systems development; a methodology that the developers can easily follow and that will cope with trust complexity and will make the development tasks visible and clear to them.

However, developing such a methodology requires an interdisciplinary research in other fields such as sociology, psychology, and social psychology. Also, there is lack of agreement among scientists regarding trust and there are a lot of definitions given that each one them reflects the discipline of the researcher.

So, we summarize the following research questions:

a) How can we define system trustworthiness?
b) What is a better way to model system trustworthiness?
c) How trustworthiness properties influence development decisions?

2 Related Work

There are a number of approaches that consider trust issues including trust modelling, reasoning, and the formation of a common vocabulary during the software development stage.

Yu and Liu [1] address the issues of trust at the requirements level of the system development process. They consider trust as a non-functional requirement, where trust is a combination of all or some quality attributes of a system under development and they demonstrate their approach by describing the behaviour of a system in the case of attack and examine defences that are needed from trust perspective.

In Yan and Cofta [2] trust is a set of statements and goals and trust domains are defined as areas of mobile communication where the trust definition is common among the various elements. Because of the subjectivity of trust definitions there are gaps between the trust domains and certain elements bridge the gap and are responsible for ensuring trust at a higher level than the one of the domains. In the methodology the entities, the domains and their interconnections are represented graphically and enable a better view of the system.

Secure Tropos [3] extends Tropos methodology with the concepts of trust, delegation, provisioning and ownership in order to allow the developer to capture trust relationships at a social and individual level.

Bimrah [4] extends the Secure Tropos [5] methodology with the concepts of request, action, trust relationship, trusting intention, reputative knowledge, recommendation and consequence in order to model trust. The developer is guided through a series of models in order to analyse and reason about trust relationships.

In [6] the proposed method makes use of the Goal Requirement Language (GRL) and Use Case Map (UCM) which both of them belong to the User Requirement Notation (URN). Specifically, trust is captured as a soft goal because of its uncertainty of whether is has been satisfied or not and because of its fuzzy nature. Further analysis of trust as a soft goal will eventually lead to well defined tasks.

The UMLtrust framework of Uddin and Zulekernine [7] considers trust from the early stages of the development process and it is a scenario-based analysis of trust. It uses UML, which is well accepted among software developers.

2.1 Limitations of Related Work

However, most of the existing approaches concentrate on a subset of parameters that affect trust and do not consider the holistic nature of trust [8]. One such parameter that is being neglected is that trust is context dependent. Moreover, the state of the art so far adopts an ad-hoc approach when tries to consider trust issues, which proves to be more difficult and costly [8]. When trust requirements are not considered in a consistent way from the early stages of the software development, but in an ad-hoc way, they will result in conflicts with the other functional and non-functional requirements of the system. So, in order to resolve these conflicts more valuable resources, such money and time, will be needed, that will still lead to a more complex solution.

In addition, developers of a project might have a different cultural background and have a different understanding of trust. Therefore, not only they need a methodology that can guide them through the software development process, but also a methodology that will establish a common understanding of trust within a technical setting among the developers. Most of the works so far focus on modelling of trust but there is not much work that has been done that will focus on the essential properties related to the trust. We believe our work contribute on this direction.

3 Proposed Approach

3.1 Trust

Trust is positive expectations of the behaviour of another actor from whom he might be positively or negatively affected [9] and plays a very important role in software systems. The actor that trusts another actor is called trustor and the actor that is being trusted is called trustee. The relationship that exists between the trustor and the trustee is called trust relationship. So, in information systems, trust between actor A and actor B is positive expectations of A about the outcome of his relationship with B, from which he might be positively or negatively affected.

There is trust between the trustor and the trustee when the trustee possesses enough characteristics that are considered signs of trustworthiness within the trustor's social context. The more such characteristics the trustee possesses the greater the trust of the trustor in the trustee will be. Moving on to the information systems, a user will consider an information system trustworthy if the information system possesses characteristics that are considered signs of trustworthiness inside the social setting.

McKnight and Chervany [10] defined four trust constructs, trusting beliefs, dispositional trust, institution based trust and trusting intention (fig. 1).

3.2 Trustworthiness

In [11] the evidence that express the trustworthiness of an actor are continuity, competence, and motivation. Continuity means that the current relationship between the two actors will continue in the future and it will not be only temporal. Competence refers to the means that the trustee possesses in order to support the trustor. An example of competence in a student teacher relationship is if the teacher has a teaching certificate. Motivation is whatever drives the trustee to support the needs of the trustor and an obvious example is if the trustor's interest matches with the trustee's interest.

In addition, McKnight and Chervany [10] define the concepts of competence, benevolence, integrity, and predictability regarding trustworthiness. Competence means that the trustee has the power or the ability to do what it is needed by the trustor, while benevolence means that the trustee also cares about the trustor and is motivated to act in his interest. Integrity means that the trustee is honest and he will keep his promise to the trustor and he will fulfil the agreement, while predictability means that the trustee's behaviour is consistent enough so that the trustor can forecast his future behaviour.

Furthermore, in [12] the properties regarding trustworthiness are called intrinsic properties that are consisted of ability, internalized norms, and benevolence. Ability is all the skills, competences, and characteristics that are required by the trustee in order to deliver in his relationship with the trustror. Internalized norms are all the principles that are considered acceptable by trustee and he behaves according to them. Finally, benevolence is when the trustee is in a relationship with a trustor as part of his own gratification and in such a relationship the trustee doesn't expect any return from the trustor.

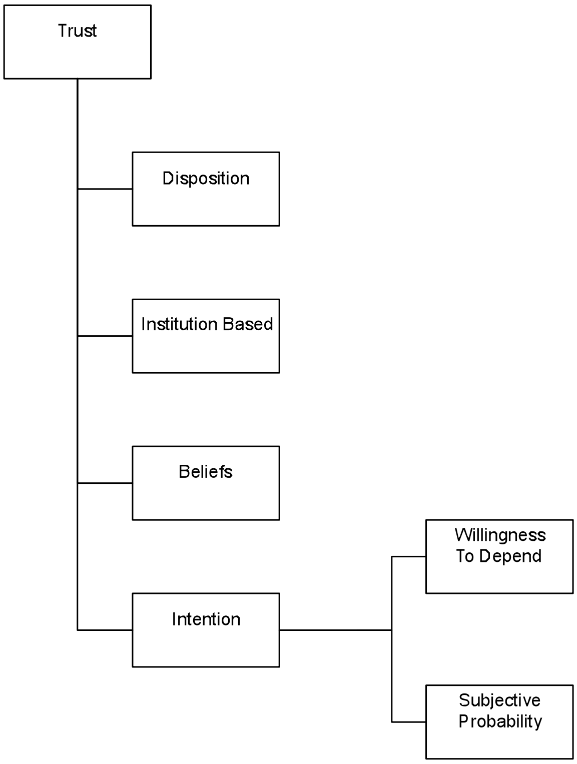

Fig. 1. Trust properties

3.3 Trust and Trustworthiness Relationship

The optimum condition though that will benefit the society is reached when there is justified trust and not when individuals show unconditional trust [13]. Justified trust is when trust of the trustor matches the trustworthiness of the trustee, and the maximum benefit occurs. Trusting less is a loss of opportunities, while on the contrary trusting too much makes you vulnerable [11].

Trustworthiness is a characteristic of a person or thing that is the object of someone's trust. If the object of our trust is worthy of that trust, then it will fulfil our expectations and our trust will be rewarded, not betrayed. If a person is trustworthy, it is considered a virtue, so if a software artefact is trustworthy, then it is considered a mark of high quality [14]. In addittion, being able to trust a software system is a prerequisite for its social acceptance [11] and when there is trust in a software product it will increase its sales and the willingness of the users to pay even more for that product [15]. Otherwise, the software system will be rejected and the development of that software system will result in a failure. Therefore, the goal is to increase as much as possible the trustworthiness of the system so that the trust of the users can be high and well placed.

3.4 Process of Trusting

Trust is not static, but dynamic, which means that it is not stable during a time period but it changes over time. When two parties engage into a relationship there are two categories of factors that have an impact on the trust process, the extrinsic and intrinsic trust factors [16]. The extrinsic trust factors are all the information about the trustee that is collected by the trustor without any direct experience, such as the trustee's reputation. On the contrary, the intrinsic trust factors are the information that the trustor collects about the trustee while having a direct experience. In the early stages of the trust relationship the extrinsic trust factors have a greater impact in the trustor's trust decision. However, as time goes by the intrinsic trust factors become more important since the trustor can make his decision on information that has been collected from direct experience.

The aim of this research is not to increase the user's trust perceptions, but to increase the actual trustworthiness of the system so that user's trust will be well placed and users will recognize this and develop trust in the system. In addition, this research is not aiming at the first stages of a trust relationship. It is concentrated on the later stages of a trust relationship, where the intrinsic trust factors have become more important, such as the ability of the system to ensure the privacy and of information.

4 Progress

A trustworthy system is a system that has the capability of meeting customer trust and the capability to meet their stated, unstated, and even unanticipated needs [17]. Moreover, trustworthiness of a software system is the assurance that the system will perform as expected [18]. Hoffman, Lawson-Jenkins and Blum [19] proposed and extended a trust model, which considers privacy, security, reliability, usability, safety, availability, and user expectations as subcomponents of trust. We consider system trustworthiness based on the following properties (fig. 2):

Privacy of personal information plays a very important role in building trust in an information system. Today, information systems have the ability to collect and store personal information very easily and by providing wider access. So, there is an increased risk for the personal information to be intentionally or unintentionally disclosed that will result in decreasing users' trust in the information system [20].

There are multiple examples in e-commerce and e-banking where the trustee is required to maintain the privacy of the customer's name, address and credit card details. Although consumers are willing to sacrifice their privacy in order to have some benefit, the privacy concerns are influencing the acceptance and adoption of new technologies [21].

Security is also of the same importance as privacy. If we don't consider trust in the design, then we might end up with a system that has security measures that are not needed and that they just make the collaboration of the users through the software system and with the system more difficult. On the other hand, security measures might not be taken in cases where it is assumed there is trust among the users or the system when it actually there in no such trust [3]. Rasmusson and Janssen [22] define two types of security, hard and soft security. Hard security is all the security mechanisms that protect the systems against any potential attack. However, there are cases that we do not only want to prevent an attack to a system resource but to protect ourselves from a provider of a harmful or low quality resource and the approach of protection in these cases is called soft security. Therefore, the receiver of the resource needs to show trust only to those resource providers that are trustworthy.

Reliability of a system can be defined as "the probability that a system will perform a specified function within prescribed limits, under given environmental conditions, for a specified time" [23]. Also, Avizienis, Laprie and Randell [18] define system reliability as the continuity of providing the correct service. The attribute of reliability of a system as a trustee contributes significantly to its trustworthiness. When a trustor assigns a subjective probability to the behaviour of the trustee this is called reliability trust and it excludes situational parameters [24].

Another additional attribute of trustworthiness is usability. According to [25] and [26] usability is:

- How easily the users learn the interface (learnability).
- The efficiency of the interface (task performance).
- How easily the users can memorize.
- The reduction of errors.
- The general satisfaction with the interface.

Research so far has shown that the usability factors have an impact on trustworthiness, especially of the websites, as they increase the perceived ability of them [27]. Moreover, usability is a prerequisite to trust, as users need to trust themselves and their ability that they can use a software system correctly [28].

Safety is when there will not be any catastrophic consequences to the users or the environment by the use of the software system [18] and maintainability is the ability of the software system to be modified or repaired [18]. Both these properties influence trust.

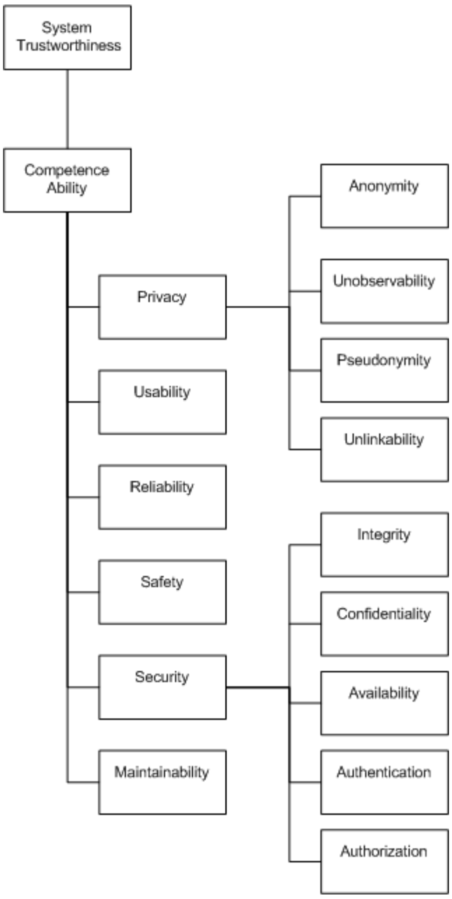

Fig. 2. System trustworthiness properties.

5 Research Contributions

The main research contribution of this work is a methodology that the developers can easily follow and will guide them towards the development of trustworthy information systems. Developers of a project might have a different cultural background and have a different understanding of trust. Therefore, not only they need a methodology that can guide them through the software development process, but also a methodology that will establish a common understanding of trust within a technical setting among the developers Also, trustworthiness is examined from a holistic perspective in order to capture all the properties that are important in developing trust and not only a subset.

Information should be communicated across trusted parties. So, in the design of the system the social architectures of trust need to be considered and reflected in the technical architectures of trust. Therefore, our goal is to capture the social relationships of trust and to verify if the design of the system conforms to them.

Furthermore, the state of the art so far adopts an ad-hoc approach when tries to consider trust issues, which proves to be more difficult and costly [8]. When trust requirements are not considered in a consistent way from the early stages of the software development, but in an ad-hoc way, they will result in conflicts with the other functional and non-functional requirements of the system. So, in order to resolve these conflicts more valuable resources, such money and time, will be needed, that will still lead to a more complex solution.

Acknowledgments. This research is under the supervision of Dr. Haralambos Mouratidis.

References

1. Yu, E., Liu, L.: Modelling Trust for System Design Using the i* Strategic Actors Framework. In: Proceedings of the International Workshop on Deception Fraud and Trust in Agent Societies, pp. 175-194. Springer (2001)
2. Yan, Z., Cofta, P.: Methodology to Bridge Different Domains of Trust in Mobile Communications. In: Proceedings of the First International iTrust Conference, pp. 211-224. Springer (2003)
3. Giorgini, P., Massaci, F., Mylopoulos, J., Zanone, N.: Requirements Engineering for Trust Management. International Journal of Information Security 5 (4), pp. 257-274 (2004)
4. Bimrah, K.K.: A Framework for Modelling Trust during Information Systems Development. PhD Thesis, University of East London (2009)
5. Mouratidis, H., Giorgini, P.: Secure Tropos: A Security-Oriented Extension of the Tropos Methodology. International Journal of Software Engineering and Knowledge Engineering 17(2), pp. 285-309 (2007)
6. Pourshahid, A., Tran, T.: Modelling Trust in E-Commerce: An Approach Based on User Requirement. In: Proceedings of the 9th International Conference on Electronic Commerce, pp. 413-422. USA (2007)
7. Uddin, M.G., Zulekernine, M.: UML-Trust: Towards Developing Trust Aware Software. In: Proceedings of the ACM Symposium on Applied Computing, pp 831-836. Brazil (2008)
8. Lo Presti, S., Butler, M., Leushel, M., Booth, C.: Holistic Trust Design of E-Services. In: Song, R., Korba, L., Yee, G. (eds.) Trust in E-Services: Technologies, Practices and Challenges, pp. 113-139. Idea Group Publishing, London (2006)
9. Mollering, G.: The Trust/Control Duality: An Integrative Perspective on Positive Expectations of Others. International Sociology 20 (3), 283-305 (2005)
10. McKnight, D.H., Chervany, L.N.: What is Trust? A Conceptual Analysis and an Interdisciplinary Model. In: Americas Conference of Information Systems, pp. 827-833. Long Beach, CA (2000)
11. Cofta, P.: Trust, Complexity and Control: Confidence in a Convergent World. John Wiley and Sons, London (2007)
12. Riegelsberger, J., Sasse, M.A., McCarthy, J.D.: The Mechanics of Trust: A Framework for Research and Design. International Journal of Human-Computer Studies 62 (3), 381-422 (2005)
13. Braynov, S., Sandholm, T.: Contracting with Uncertain Level of Trust. Journal of Computational Intelligence 18 (4), pp. 501-514 (2002)

14.Miller, W.K., Voas, J.: The Metaphysics of Software Trust. IT Professional IEEE Computer Society. pp. 52-55 (2009)

15.Masthoff, J.: Computanionally Modelling Trust: An Exploration. In: Proceedings of the SociUM Workshop assosiated with the User Modelling Conference. Greece (2007)

16.Josang, A., Keser, C., Dimitrakos, T.: Can we manage Trust? In: Hermann, P., Issarny, V., Shiu, S. (eds.) Trust Management. LNCS, vol. 3477, pp. 13-29. Springer, Berlin (2005)

17.Jayaswal, K.B., Patton, C.P.: Design for Trustworthy Software: Tools, Techniques, and Methodology of Developing Robust Software. Prentice Hal, London (2007)

18.Avizienis, A., Laprie, J.C., Randell, B.: Dependability and its Threats: A Taxonomy. In: 18th IFIP, pp. 91-120. Kluwer Academic Publishers (2004)

19.Hoffman, J.L., Lawson-Jenkins, K., Blum, J.: Trust Beyond Security: An Expanded Trust Model. Communications of the ACM 49 (7), pp. 95-101 (2006)

20.Rohm, J.A., Milne, R.G.: Just what the Doctor Ordered: The Role of Information Sensitivity and Trust in Reducing Medical Information Privacy Concern. Journal of Business Research 57, pp. 1000-1011 (2004)

21.Spiekermann, S., Cranor, L.F.: Engineering Privacy. IEEE Transactions on Software Engineering 35 (1), pp. 67-82 (2009)

22.Rasmusson, L., Janssen, S.: Simulated Social Control for Secure Internet Commerce. In: Proceedings of the 1996 New Security Paradigms Workshop. ACM (1996)

23.Stapelberg, F.R.: Handbook of Reliability, Availability, Maintainability, and Safety in Engineering Design. Springer, London (2009)

24.Josang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support Systems, 43, pp. 618-644 (2007)

25 Nielsen, J.: Usability Engineering. Academic Press, London (1993)

26. Shackel, B.: Applied Ergonomics Handbook. IPC Science and Technology Press, Guilford (1975)

27.Roy, C.M., Dewit, O., Aubert, A.B.: The Impact of Interface Usability on Trust in Web Retailers. Internet Research: Electronic Networking Applications and Policy 11 (5), pp. 388-398 (2001)

28.Sasse, M.A.: Usability and Trust in Information Systems. In: Mansell, R., Collin, S.B. (eds.) Trust and Crime in Information Societies, pp. 319-348. Edward Elgar (2004)