Intangible Trust Requirements - How to Fill the Requirements Trust "Gap"?

Tim French
Department of Computer Science & Technology
University of Bedfordshire
Park Square Campus, Luton LU1 3JU, UK.
Tim.french@beds.ac.uk

Wei Huang
Department of Computer Science & Technology
University of Bedfordshire
Park Square Campus, Luton LU1 3JU, UK.
Wei.Huang@beds.ac.uk

ABSTRACT

Previous research efforts have been expended in terms of the capture and subsequent instantiation of "soft" trust requirements that relate to HCI usability concerns or in relation to "hard" tangible security requirements that primarily relate to security assurance and security protocols. Little direct focus has been paid to managing intangible trust related requirements *per se*. This 'gap' is perhaps most evident in the public B2C (Business to Consumer) E-Systems we all use on a daily basis. Some speculative suggestions are made as to how to fill the 'gap'. *Visual card sorting* is suggested as a suitable evaluative tool; whilst *deontic logic trust norms* and UML extended notation are the suggested (methodologically invariant) means by which software development teams can perhaps more fully capture hence visualize intangible trust requirements.

Author Keywords

Intangible trust requirements; visual card-sorts; deontic norms;

ACM Classification Keywords

H5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous.

INTRODUCTION

The extant trust related research literature is vast and highly diverse. A full literature review of intangible trust in B2C e-service contexts lies outside the scope of this exploratory paper. The interested reader is directed to both [1] and [2] for a comprehensive literature review. One of the difficulties is that the notion of trust is closely related to other concepts such as reliance, competence, trustworthiness and credibility. Deutsch [3] was one of the first modern writers to seek to build a formal model of trust. He defined trust in terms of an individual confronted with an ambiguous path. Further, the path may lead to either an event leading to a beneficial outcome (Va+) to that individual or to an event perceived as being harmful (Va-). This individual perceives that the occurrence of Va+ or Va- is dependent on the behaviour of another human agent. Finally, the strength of Va- is greater than the strength of Va+. Essentially, his view of trust is of a trust relationship in which events are linked to other events, each of which has beneficial or non-beneficial paths. For a trust relationship to occur, the harmful path is more significant than the beneficial path. Risk is an essential property of the environment within which a choice of paths occurs.

The notion that trust building between individuals takes place within information spaces that are both potentially risky to the participants and where incomplete information is available to the human actors has been widely accepted and developed by many subsequent researchers [4]. Within this information space the notion of expectation is central to many writers. For example, Gambetta [5] provides us with a rich and potentially computationally useful definition that encapsulates the notion of trust as expectation: 'Trust (or symmetrically, distrust) is a particular level of 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 or his capacity ever to be able to monitor it) and in a context in which it affects his own action'. This idea of trust as an expectation (either rational or affective or a mixture of both) is a closely related concept to that of confidence levels. Confidence can be defined as a conscious or unconscious act or mental state involving the placing confidence in something or someone. The idea that this confidence level can be formalized, hence measured is developed by Marsh in his highly influential PhD thesis [6]. Human actors often invest their trust in a particular object of trust. This object may be another human agent or some artifact (such as a B2C Web-site). Indeed, within our intended scope of enquiry a company web-site acts as a central focal point within an information space (cyberspace) within which actors can choose either to follow "selfish self-interested" paths or act in the interests of others ("benevolent" paths). With the notion of an object of trust in mind, researchers have attempted to explore and develop the notion of trust as credibility, both within off-line and on-line contexts [2]. Kim *et al.*, [2] have variously identified the sub-components of the credibility of an object of trust as comprising: honesty, expertise, predictability and reputation. Research appears to indicate that *credibility* is driven by the behavioural predictability of a trusted object, for example a web-site [2]. Indeed, many researchers, such as those of [7], define trust as comprising the dimensions of trustworthiness and expertise. These dimensions are closely related to the notion of credibility. Many have stressed the importance of trust building over time (i.e. the temporal dimension of trust). Thus the concept of trust as *reputation* has been developed by those seeking to quantify and measure high-level organisational trustworthiness in businesses and organizations, and most recently in Virtual Organisations [8]. Trust is not only dependant on our past experiences but also on an expectation of reliability and confidence in future events too. These aspects have been incorporated into various formal and informal models of trust building in relation to specific methodologies, such as agile [9-10]. An important aspect of trust building is the degree to which *affective* vs. *rational* components are involved. It is clear from the literature that the affective component has been relatively under researched in comparison to the rational cognitive dimension [1].

TRUST and UX

Hassenzahl and Tractinsky, [11] deem that UX is influenced by a user’s internal state (e.g. predispositions, expectations, needs, motivation, affective state), as well as by the characteristics of the designed system (e.g. complexity, purpose) as well as by the environmental context within which the interaction occurs (e.g. organisational, social and cultural setting, and perceived meaningfulness of the activity). With respect to B2C e-shopping user contexts, reinforcing initial user trust building expectations is critical if businesses are to leverage full value from their B2C sites. B2C trust building via the HCI (Human Computer Interface) layer has been extensively studied and many of the trust building factors have been identified. Trust signs, seals, physical address details, rich media *et al.,* all serve to act as trust builders both as affective and rational trust drivers. A review of the relevant literature and a summary of trust determinants can be found in [12]. There is little direct methodological support to B2C system developers for intangible trust requirements capture. Rather, implicitly such systems are enhanced with respect to trust only much later via acceptance testing and usability testing, if at all. Thus, intangible trust requirements in sharp contrast with tangible security requirements are not typically captured early enough within the software development lifecycle.

One of us has successfully used visual card sorts in the context of an SME to improve the UX of their B2C web-site. The method has proven to be useful in revealing to both developer and end-users semi-tacit trust knowledge. Card sorts are not of course a new technique but their use in the context of validating intangible trust designs of for instance B2C e-banking is relatively new and may hold promise for the future [13]. The main advantage is that a cognitive "map" can be created from early (even paper based) web-site visual designs that reflects both the tacit and semi-tacit knowledge of site users. This "map" can aid in probing customer perceptions of UX, including affective and rational trust responses to various visual design features. Card-sorting can also be used to validate early paper-based prototype designs as well as later in the release cycle so as to probe "live" or pre-release user trust site perceptions.

The final site design is however, ultimately only the product of earlier application of methods, tools and requirements gathering activities. Indeed, we go on to argue below that there is an intangible requirements 'gap' within existing methodologies. That is, capture of intangible requirements is often implicit and does not always form an integral part of the normative team design process that ultimately leads to a given trusted or distrusted UX.

A METHODOLOGICAL TRUST 'GAP'?

The familiar software development lifecycle methods used in industry such as agile approaches, XP (Extreme Programming), RAD (Rapid Applications Development) and indeed the typical waterfall models all tackle the issue of requirements in different ways. Agile emphasises face-to-face interactions and iterative development within time-boxes, in which case it may be expected that some informal stakeholder trust expectations may exist during the lifecycle. As a type of agile software development, XP aims to improve responsiveness for requirement changes during the software lifecycle and it normally has multiple short development cycles instead of one long one in order to reduce the cost of changes. Unlike typical waterfall models, which can be treated as “plan-driven” or “predictive” so trust is relatively easier to be built up, agile methods, which have much in common with XP and RAD, are “adaptive” and can respond quickly to requirement changes. Thus trust plays an important role in this kind of “adaptive” method [9]. The following points seek to summarise and compare a number of typical lifecycle methods to see whether, when and how intangible trust issues are considered:

- *Waterfall*: Intangible trust requirements are typically embedded within feasibility study, and in a requirements catalogue as non-functional requirements explicitly agreed by client and developer;
- *Agile (e.g. XP)*: Trust is vital at every stage among developer team members. More specifically, trust is tested during the meetings and at the end of each and every iteration. Trust is built up due to its iterative nature and its primary focus upon interpersonal trust in the development process. Tested software is generated at the end of every iteration and this helps to build a sense of credibility (if the tested software is working as scheduled!);

- *RAD (Rapid Applications Development)*: Often implicit and embedded within evolving software artifact itself (all stages) Trust is most likely to emerge as an "issue" via acceptance testing and system walkthroughs once artifact is well refined. "Look and feel" of software engenders customers to engage through the use of branding, metaphor and narrative;
- *Test-driven development (TDD):* Not widely considered? Some signs of confidence / trust build up when all test cases "pass" (which only may mean that the code meets all the defined/explicit requirements. (Trust is perhaps not widely considered explicitly here partly because TDD is a relatively new technique.)

Ideally trust building aspects should be initiated at the very beginning among all system stakeholders (i.e. at the requirements stage) no matter which lifecycle method is used. Trust requirements are important to final users as well as to other stakeholders such as developers, managers, clients, and system sponsors. Indeed many consider trust amongst Agile teams, tools and techniques for example to be absolutely vital to help generate a credible "win-win" specification. Yet as many industry practitioners acknowledge this is rarely the case in practice due to deep seated cultural differences as between developers and their clients.

It would appear that there is a methodological 'gap' with respect to intangible trust requirements, particularly with respect to aspects of UX that encompass hedonic, emotional aspects - not merely trust as rational decision making and tangible security. There is a methodological trust “gap”, particularly at the requirements stage. None of the notations within the UML (Unified Modelling Language) for example directly support intangible trust requirements - aside from generic use-case and domain models. Rather, the main focus is upon tangible security and assurance aspects. Methods such as MEASUR [10] claim to add value to existing approaches via ontology charting by seeking to capture not only semantic entities and their relationships but also organisational contexts and culture, including normative methods of working, both formal and informal. This may perhaps serve to reveal implicit trust aspects within workgroups or indeed trust expectations concerning the presentation layer of the system. However, MEASUR is not in fact as yet widely used outside academia, despite many years of effort. This lack of adoption limits its potential impact and relevance to addressing the trust gap, despite recent efforts to formalize, align and integrate MEASUR with modern component based design principles [10].

HOW TO FILL THE TRUST "GAP"?

Dyadic trust between an e-service provider and consumer (trust as a set of expectations as to future behaviour, reliability, service quality *et al.*,) is typically influenced by both rational and affective drivers that in turn serve to influence technology acceptance levels. The well known and heavily cited TAM (Technology Acceptance Model) of a type proposed by [2] exemplifies this vision. It is implicit within such models, that trust is intimately related to risk; that is to say where there is no risk trust is not relevant. Rather, the higher the risk factors the more reliance needs to be placed by stakeholders on intangible trust requirements so as to mitigate perceived risk.

However, it would be a mistake to say that mere security assurance equates to trust *per se*; rather, intangible perceptions of trust create the necessary pre-conditions and set of constraints within which systems are procured, developed, and are ultimately released. Thus, trust is acts as a super-set (universe of discourse) within which tangible security assurance standards are seen to operate. However, it is important to note that *mere* security assurance itself (e.g. secure message protocols, cryptographic techniques) can of themselves *never fully meet* the intangible trust concerns of users as part of their UX. For one thing, wider intangible cultural trust norms differ greatly across and within societies and cultures [12] and thus are highly relevant to "shared meanings" across cross-trans/national software development teams working across borders or with culturally specific B2B partnerships. Although intangible trust perceptions/expectations between business partners or as between developers and clients mediate requirements negotiations from the earliest stage of the software development lifecycle, there are *two* fundamental problems that we seek to address:

a) Firstly, intangible trust requirements are often at least partially *implicit*, hence developers and clients may not themselves fully realise its potential impact upon eventual system acceptance until too late in the development of the system. When (due perhaps to disagreements or ambiguity or un-stated sets of trust expectations, i.e. norms and meta-norms) such issues may become more explicit, resulting in a mismatch as between a norm and an agreed set of functional requirements. Such matters cannot even be realised (hence alterations made) by system developers or their clients before actual release of the system unless intangible trust is fully and richly articulated (hence made fully visible) to all system stakeholders.

Candidate solution?: Deontic logic has previously been used to define norms and meta-norms in the context of enabling MONA (Portugese Acronym for a norm modeler for tailorable user-interfaces) [14]. It may be that in the future the definition of high-level trust specific norms and meta-norms can (since the natural language version of deontic logic is easily interpretable by clients) be potentially useful in framing intangible trust issues - thus potentially impacting on the design of a user's UX. A potential advantage of the use of deontic norms is that they are expressive enough to reflect well known cultural differences of the wider social world within which the system is seen to operate.

b) Secondly, there is a *notation gap* with respect to articulating intangible trust requirements. Within the rich and expressive notational vocabulary of the UML, there is no specific notational support for trust, other than as a natural text narrative to enrich the domain model. Various notations and formalisms such as state-charts have been adapted to reflect some aspects of intangible HCI trust requirements. However these extended or otherwise specially adapted charting methods are not widely used outside academia. Formal and mathematical notations claim to have been used for trust, yet they often only actually reflect tangible security paradigms. There has been some emergent work on the development of trust specific notations and methodologies such as *The Shared Meanings Design Framework* (SMDF) to capture trust requirements across stakeholder groups. Few if any of these notations are used outside academia.

Candidate solutions?: Perhaps suitable extensions to the well known UML notation can be provided to support the explicit articulation of intangible trust issues (for example the domain model). As yet, this potential has only been tentatively explored in relation to intangible trust [15], though recently approaches such as UMLTrust seek to offer support for intangible as well as tangible security aspects: trust policies, scenarios as well as trust certification [16]. One alternative path going forward is perhaps that one of the many trust specific notations to emerge out of academia will be adopted or otherwise influence industrial practitioners, such as the SULTAN (Simple Universal Logic-oriented Trust Analysis Notation) [17] and associated tool-kit previously developed at UCL.

In our earlier discussions it was apparent that there is no one methodology that is universally adopted. Rather, methods are selected by client-developer partnerships according to the "best fit" to whatever type of software system is proposed. For this reason we are very hesitant to supply any definitive answer to the trust "gap" across every method; but the above suggestions may perhaps at least prove useful as potential candidate solutions. In any event it is our contention that card-sorts have been shown to add value to the probing of intangible trust perceptions using B2C sites. Either in relation to early designs or in relation to pre- or post release UX intangible trust evaluation. So whilst various possibilities exist with respect to enriching existing methodological practice, UX trust perceptions can at least be probed empirically, once an artifact has emerged from the development team. It would of course be more desirable if as part of whatever methodology is used to develop artifacts, that intangible trust notational support could be agreed upon and more widely adopted. Thus far, whilst various tentative suggestions have been made by academia, industrial practice has tended only to support tangible security requirements at the expense of intangible trust concerns.

CONCLUSION

Intangible trust forms an important yet somewhat elusive part of both UX, and wider technology acceptance. Without trust building (explicit or implicit or both) systems will simply be not adopted or "work around's" will be employed. Despite the importance of intangible trust building, there appears to be a methodological and notation 'gap'. The framing of system design within explicit trust norms may prove to be useful since any methodology could be "front-ended" by a set of norms that are method and notational independent of any existing method. The alternative or complementary approach is to leverage an existing notation (e.g. the UML) for intangible trust or perhaps (even more speculatively) to seek to influence industry standards and methods such that they fully encompass intangible trust requirements. Others have tried to develop their own methods yet these are surely doomed to failure unless industrial developers and their clients feel that the trust gap is worth filling (adds real "bottom line" value?) Perhaps at present there is a certain cynicism that leads to rapid system release followed by numerous "patches" that seek to paper-over gaps in requirements. This is both the fault of clients (too demanding time-scales) and developers. But it is also because of a "gap" in the industry methods and notations currently deployed.

As wider notions of UX grow it is to be hoped that this "gap" will be filled as all stakeholders come to realize the importance but also acknowledge the intractability of trust; including the fact that we lack models of trust that take into account "obvious" cultural differences. Thus, there will be no quick "fix", rather the trust gap reflects deep seated cultural divide as between system stakeholders, organisational needs and current paradigms.

Many challenges remain not the least of which is: how to define intangible trust in the first place. Deciding how and what to "measure" becomes a central question - particularly perhaps with respect to the impact of cross-cultural trust norms. The extant literature in B2C e-trust has perhaps tended to place over-relied on methods such as questionnaires and under developed ways of probing user cognition such as card-sorts. Perhaps in the future, hybrid approaches that seek to triangulate as between physiological metrics and cognitive metrics by incorporating neuroscience may add value to evaluating trust as part of UX [1]. This may in turn lead to the definition of objective physiological trust metrics as well as subjective metrics in relation to the UX.

If industry is willing to embrace new "blue-sky" techniques and sees added value in funding studies in Usability Labs., as part of their requirements gathering /interface design validation studies then intangible trust requirements gathering activities could form a normative part of *every* software project that has end users irrespective of the actual choice of methodology. As yet though, too often failure to address intangible trust perceptions result in lack of adoption or expensive "fixes" and software re-releases.

To cite one well known UK Public Sector instance: the lengthy adoption of the NHS (National Health Service) GP-to-hospital "*Choose and Book*" specialist referral e-booking system has been frequently ascribed as being due to an initial failure to address stakeholder trust and mistrust issues at an early stage in the system's initial specification. This initial failure led to not only to an initial lack of adoption by GP's, but was the prime cause of numerous subsequent system upgrades over a lengthy eight year time scale [18].

REFERENCES

[1] French, T. Towards an E-Trust Framework: trust as a semiotic phenomena. PhD Thesis, (2009). Reading University, School of Systems Engineering, UK.

[2] Kim, D.J., Ferrin, D.L., Rao, H.R. A trust-based consumer decision-making model in electronic commerce: The role of trust, perceived risk, and their antecedents. *Decision Support Systems, 44* (2008) 544–564.

[3] Deutsch, M. Trust and suspicion. *Journal of Conflict Resolution, 2(3)* (1958), 265-279.

[4] Corritore, C., Krasher, B., & Wiedenbeck. On-line trust: concepts, evolving themes, a model. International *Journal of Human-Computer Studies, 58,* (2003) 737-758.

[5] Gambetta, D. (Ed.) *Trust: Making and Breaking Cooperative Relations.* (2000) Oxford: Basil Blackwell Publications.

[6] Marsh, S.P. Formalising Trust as a Computational Concept. PhD Thesis, (1994). Stirling University, UK.

[7] Fogg, B.J., Marshall, J., Laraki, O., Osipovitch, A., Varma, C., Fang, M., Paul, J.,Rangnekar, A., Shon, J., Swani, P. and Trienen, M. What Makes a Website Credible? : a Report on a Large Quantitative Study. *Proc. SIGCHI conference on Human factors in Computing Systems,* (2001) 61-68.

[8] French, T. Collaborative Virtual Organisation Trust Measurement: Leveraging Corporate Governance Metrics, *Procs. IEEE, i-Society*, (2010) 502-509.

[9] Hasnain, E, and Hall, T. Investigating the Role of Trust in Agile Methods using a Light Weight Systematic Literature Review. *Procs. 9th International Conference on Agile Processes in Software Engineering and Extreme Programming*. Limerick, Ireland, Jun 10-14, 2008. In: *Lecture Notes in Business Information Processing, Volume: 9,* (2008) 204-207.

[10] Poernomo, I., and Tsaramiris, G. Ontology Based UML2 Architecture Generation. *Procs. ICISO,* (2010) 314-321.

[11] Hassenzahl, M, Tractinsky N. (2006). User experience –a research agenda. *Behaviour & Information Technology*, 25 (2), March-April 2006, 91-97.

[12] Wang, Y.D., and Emurian, H.H. An Overview of online Trust: concepts, elements and implications. *Computers in Human Behaviour, 21*, (2005) 105-125.

[13] French, T., and Opatola, K. A Pilot Investigation of E-Banking trust perceptions amongst the Yoruba and Igbo of Nigeria. *Procs. IWIPS 2010,* (2010) 107-116.

[14] Neris, V.P., and Baranauskas. User Interface Design Informed by Affordances and Norms Concepts. *Procs. ICISCO,* (2010) 133-140.

[15] Oussena, S. and French, T. Integrating the Semiotic into UML via Enhancing and Cross-validating Use Case with an Enriched Domain Model. *International Journal of Sociotechnology and Knowledge Development (2010),1,3,*15-31.

[16] Uddin, M., and Zulkernine, M. UMLTRust: Towards Developing Trust-Aware Software, *Procs. of the 2008 Symposium on Applied Computing,831-836,* New-York, ACM.

[17] Grandison, T. Trust management for internet applications, PhD Thesis, (2002). University of London.

[18] *Referral Management: Lessons for Success*. King's Fund, 2010. ISBN 978-1-85717-600-1