Combining Risk-Management and Computational Approaches for Trustworthiness Evaluation of Socio-Technical Systems

Nazila Gol Mohammadi[1], Torsten Bandyszak[1], Abigail Goldsteen[2], Costas Kalogiros[3], Thorsten Weyer[1], Micha Moffie[2], Bassem Nasser[4] and Mike Surridge[4]

[1]paluno - The Ruhr Institute for Software Technology, University of Duisburg-Essen, Germany
{nazila.golmohammadi, torsten.bandyszak, thorsten.weyer}@paluno.uni-due.de
[2]IBM Research Haifa, Israel
{abigailt, moffie}@il.ibm.com
[3]Athens University of Economics and Business, Athens, Greece
ckalog@aueb.gr
[4]IT-Innovation Center, School of Electronics and Computer Science, University of Southampton, Southampton, United Kingdom
{bmn, ms}@it-innovation.soton.ac.uk

Abstract. The analysis of existing software evaluation techniques reveals the need for evidence-based evaluation of systems' trustworthiness. This paper aims at evaluating trustworthiness of socio-technical systems during design-time. Our approach combines two existing evaluation techniques: a computational approach and a risk management approach. The risk-based approach identifies threats to trustworthiness on an abstract level. Computational approaches are applied to evaluate the expected end-to-end system trustworthiness in terms of different trustworthiness metrics on a concrete asset instance level. Our hybrid approach, along with a complementary tool prototype, support the assessment of risks related to trustworthiness as well as the evaluation of a system with regard to trustworthiness requirements. The result of the evaluation can be used as evidence when comparing different system configurations.

Keywords: Asset Modelling, Socio-Technical-System, Computational Evaluation, Trustworthiness Attributes, Metrics, Risk Analysis, Evaluation

1 Introduction

The technological settings in which information systems are designed, developed, and deployed have drastically changed with the advance of new technologies such as cloud computing. Socio-Technical Systems (STS) are often software-intensive information systems that interact with a variety of other software systems, as well as humans and physical entities [1, 2]. To enable designing systems with higher trustworthiness, it is crucial to analytically evaluate and estimate the trustworthiness of a system with a thorough analysis of risks and mitigation actions. This includes identifying

controls to prevent threat activity at run-time. The results of this analysis should be addressed appropriately in subsequent development phases. For instance, threats identified early in the system design could yield certain trustworthiness requirements that may guide the selection or re-use of components or services. Explicit documentation of design decisions that affect the trustworthiness of a system is also essential. STS designers may need guidance when deciding whether including a certain mitigation mechanism in the system will result in an actual increase in trustworthiness, and eventually pay off. Therefore, evaluation of the End-to-End (E2E) trustworthiness of different system configurations can give some confidence in whether the detected threats are indeed prevented. Despite a large amount of literature addressing trustworthiness evaluation, the E2E evaluation of multi-faceted trustworthiness remains an open research problem. Some approaches merely focus on reliability [3] or security [4]. In contrast, our evaluation approach is based on a holistic taxonomy of software quality attributes and metrics that contribute to trustworthiness [15], including compliance, privacy, usability, complexity and many more.

We employ two complementary techniques: risk-based and computational analysis. The risk-based approach is applied at very early stage on an abstract model of the system which is independent of concrete component realizations. Complimentarily, the computational approach is performed at a later stage, on a more concrete level. It uses trustworthiness metrics, and involves calculating and aggregating them according to the complex system structures, to produce E2E metric values. Our proposed hybrid approach is a comprehensive evaluation approach that is applicable on many levels of granularity. To the best of our knowledge, no existing approaches combine these two techniques such that the consideration of threats and potential mitigations are evaluated once concrete assets are available and composed to build the system. We focus particularly on software assets that are accessible via an online marketplace with certificates holding trustworthiness metric values. Previous work aimed at establishing such a software marketplace to allow designers to select trustworthy system assets based on certificates and to compose them to create a new system [6].Our hybrid approach allows designers to evaluate a system's trustworthiness and make good design choices based on risk assessment and trustworthiness metrics. The tool support also aids designers by automatically generating software requirement documents and trustworthiness reports as evidence. The information in these documents is organized in a hierarchical way, allowing expert users to drill-down to the desired level of detail.

The remainder of the paper is structured as follows: In Section 2, we present the background and give a brief overview of existing techniques for evaluating trustworthiness. Section 3 presents our approach and evaluates it using an application example from the Ambient Assisted Living (AAL) domain. Section 4 summarizes our work.

2 Background and Related Work

This section presents the background and fundamental concepts used in our paper. We base our work on trustworthiness attributes that aim to manifest the trust concerns of end-users in an objective way. Trustworthiness attributes are quantified using trust-

worthiness metrics, that measure system (or individual components) behaviour, based on raw measurements, i.e., observable system properties.

Design-time models allow domain expertise to be encoded and reused in a systems' design. Such models may have different levels of abstraction. For example, an abstract system model can be used to help system designers to graphically identify and analyse the threats that can arise in a system [2], before knowing the actual deployment details. In STS, there is a particular need to explicitly specify the dependencies between assets in the system (e.g., host of application). An Asset is anything of value in an STS [7, 8], including software, hardware and humans. A Workflow model specifies a set of concrete asset instances, as well as their interrelations. A system can be described using several such workflows, each representing a certain process (or use case) performed by the system.

There are many standards and methods [9, 10, 11, 12] that describe the risk management methodology and provide support in the process of identifying the system assets and relevant threats. The CORAS project [13] aimed at simplifying the task using a graphical approach to identify, explain and document security threats and risk scenarios. However, these approaches depend on humans. Microsoft's SDL threat modelling tool provides a graphical user interface for developers to generate threat models based on software architecture diagrams. However, Microsoft's SDL threat modelling tools may be more likely to overlook threats beyond the scope of STRIDE [14] (e.g., human-centred attacks) unless they also involve security professionals.

The need to evaluate the overall trustworthiness of a system has been recognised by several researchers. Elshaafi et al. [15] present an approach towards measuring the trustworthiness of a service composition focusing on run-time monitoring and targeting reputation, reliability, and security considering several service compositions. Similarly, Zhao et al., [16] propose a framework for trustworthy web service management, which aggregates the availability, reliability, and response time of services composed in sequence, parallel, conditional, and loop structures. Other approaches, such as [17], focus on reputation only by aggregating service ratings in order to determine the provider's trustworthiness. Cardoso et al., [18] utilize graph reduction mechanisms and respective formulas for aggregating time, cost, and reliability of service workflows. Hwang et al. [19] propose a probabilistic approach for estimating certain quality of service of respective compositions. While the above approaches support a large number of composition patterns, they focus on a limited set of trustworthiness metrics. Closer to our approach is the work of Jaeger et al. [20] where they support a wider set of QoS metrics.

3 Trustworthiness Evaluation of Socio-Technical Systems

This section describes our trustworthiness evaluation approach. First, a conceptual model of our approach is presented. Second, we describe how we combine two existing techniques at two different abstraction levels.

3.1 Overview of Our Approach

Meta-model for Design-Time Trustworthiness Evaluation. The fundamental concepts and their relations are depicted in Fig. 1.

We distinguish between *Asset Categories*, generic types of system building blocks on an abstract level, and *Asset Instances*, concrete instances pertaining to a certain category, e.g., a concrete software application or implementation. A *Threat* is a situation or event that, if active, could undermine the value of an asset by altering its behaviour. *Controls* are trustworthiness requirements that aim at mitigating threats.

For computational evaluation of trustworthiness in our approach, *Metrics* are used as functions to quantify system trustworthiness. A Metric is a standard way for measuring and quantifying certain trustworthiness attributes and more concrete quality properties of a system [5]. Metric values of a specific Asset instance are provided within a trustworthiness certificate that is often provided together with the software itself on a marketplace.

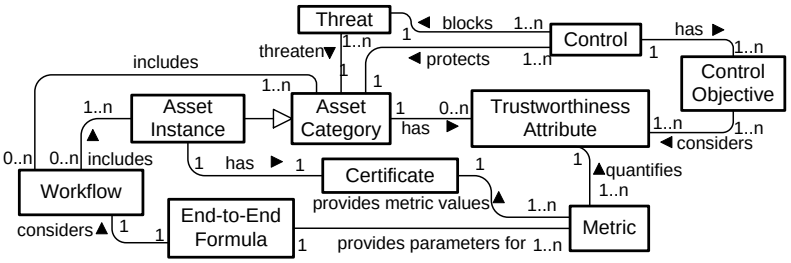

Fig. 1. Conceptual Model for Trustworthiness Evaluation

A *Certificate* is created by a certification authority that evaluates a software system or asset in order to confirm that it meets some trustworthiness goals. It describes all observed trustworthiness properties of the software, as well as related evidence in terms of certified metric values. In order to enable trustworthiness computation for a whole *E2E* system configuration, and thereby aggregate the certified metric values for each of its asset instances, E2E formulas are required. Since *Workflows* describe the concrete instance relations, each E2E formula is particular to a certain Workflow.

Combining Risk Assessment with Computational Approaches toward Trustworthiness Evaluation. In the proposed E2E trustworthiness evaluation, the risk-based approach is performed on the level of asset categories while the trustworthiness metric computation is based on metric values of asset instances (illustrated in Fig. 2). Trustworthiness evaluation starts with a design-time system model which describes the general building blocks of an envisioned STS in an abstract level. It includes both physical, and logical assets (e.g., software), as well as humans that interact with the system. This model is independent of concrete realizations, i.e., without considering which asset instances that shall be deployed as implementations of software assets.

At this early stage our approach already allows us to identify threats to correct system behaviour at run-time. To this end, a knowledge base of threats is used to identify relevant threats to the asset based on its type (e.g., logical asset, physical asset, etc.) and its relations patterns (e.g., client-server relation). Given the threats and potential

controls the designer is provided with a statement on the risks and corresponding actions that can be taken or at least planned for at design-time. Based on this information, asset structures may be revised, or informed decisions on the concrete asset instances can be made.

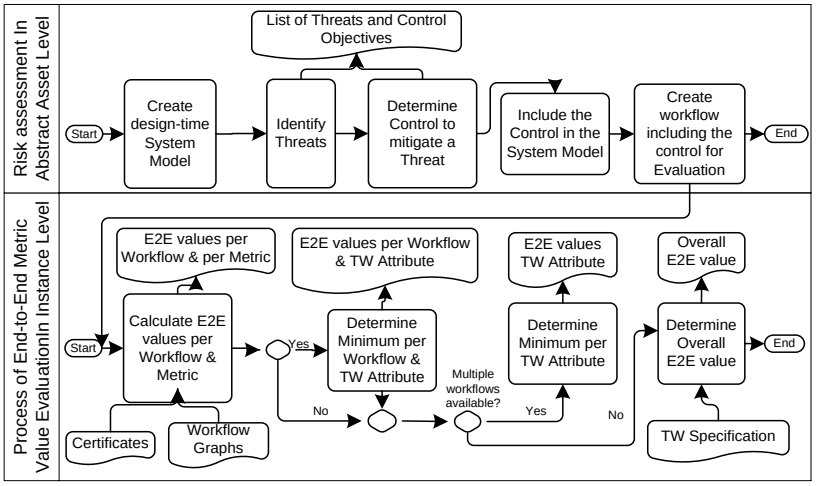

Fig. 2. Trustworthiness evaluation in two different abstraction layers

Once concrete instances (e.g., available on a marketplace) of the asset categories are selected and modelled in terms of one or multiple workflows, the designer can then use the computational trustworthiness evaluation approach to calculate E2E trustworthiness based on the certified metric values of each of the asset instances, and the E2E formulas for each relevant workflow.

The system structure needs to be considered and reflected in the E2E value. We considered different component structures for determining an "E2E" trustworthiness value based on metrics. Redundancy structures, which are defined as a means to assure correct system performance and thereby increase trustworthiness levels, were especially considered in the E2E trustworthiness calculation. The explicit description of respective metrics is a precondition for the calculation of E2E trustworthiness value, which requires certified metric values of each involved asset as parameters.

The resulting E2E trustworthiness values provide detailed information about the aggregated trustworthiness of the asset instances that are involved in a certain workflow. This allows the designer to relate the threats to the affecting trustworthiness values, and also evaluate and substantiate the effectiveness of applied design-time controls. Assuming that the metric values reflect the existence of controls (e.g., confidentiality, authentication), then the quantification enhances the evaluation. In order to facilitate the interpretation of the calculated E2E trustworthiness, the initial values that are particular to a certain workflow can again be aggregated so that finally one value for the whole system trustworthiness can be obtained. To this end, for instance, a pessimistic approach can be followed, and the minimum of two or more metrics per attribute, or workflows can be calculated. A required precondition is that the value ranges of different metrics are comparable.

3.2 Application Example

This section describes an application example for demonstrating our two-fold approach on different levels of abstraction (illustrated in Fig. 5).
The example scenario focuses on a Fall Management System (FMS) from the AAL domain. FMS allows elderly people in their homes to call for help in case of emergency situations. These emergency incidents are reported to an Alarm Call centre that, in turn, reacts by e.g., dispatching ambulances or other medical caregivers, e.g. the relatives. The starting point for evaluating the trustworthiness of such an STS is a design-time system model (depicted in Fig. 3). An elderly uses a *Personal Emergency Response System* (PERS) device to call for help, which is then reported to the Alarm Call Center that uses an *Emergency Monitoring and Handling Tool* (EMHT) to visualize, organize, and manage incidents. Hence, the EMHT is a software service hosted by the Alarm Call Center operated by a Healthcare authority. Emergency notification and Ambulances Services, which are run on mobile phones of relatives, or by Ambulance Stations respectively, are called in order to require caregivers to provide help.

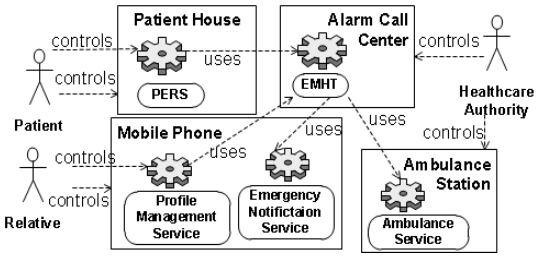

Fig. 3. Design time System Model of the Fall Management System Specifying Asset Categories

Identification of Threats and Controls. Based on the design-time system model, which specifies the relevant asset types of the system, and their relations, the trustworthiness evaluation is first performed on this abstraction layer of abstract assets. The "Evaluate System E2E Trustworthiness" use case of our prototype tool supports the designer in this task. The model file is passed to the System Analyser for analysing the threats that may affect each system asset. The System Analyser reports for each asset type the related threats as well as the potential controls that can be applied to prevent or mitigate the threats. For instance, a threat that may arise at run-time is the unavailability of the EMHT asset. This will probably lead to a failure of the whole STS, since the EMHT is a central service that enables and facilitates handling incoming calls for help. Hence, a possible control to react to this threat at run-time may be service substitution, i.e., to switch to another backup service that may also be a different implementation of the asset category. In order to address and implement this control at design-time, two or more concrete EMHT realizations have to be considered as redundant instances of the EMHT asset. The identified control will be considered (including redundant asset instances) and modelled as a workflow.

Trustworthiness Calculation for Asset Instance Configurations. As a required precondition for E2E trustworthiness calculation, the designer has to select the evaluation criteria to be used, i.e., the weights of relevant trustworthiness attributes to the overall E2E TW. The weights represent the designer's preferences regarding the

relevance of each trustworthiness attribute. The designer continues by uploading one or more workflows. Fig. 4 shows an exemplary workflow for our FMS example. The designer specifies the redundant asset instances EMHT_1 and EMHT_2 and Ambulance_Service_1 to 3 for the asset types "EMHT" and "Ambulance_Service" respectively. Based on the Workflow graphs, the Formula Builder of the E2E TWE tool will create formula skeletons for any kind of metrics.

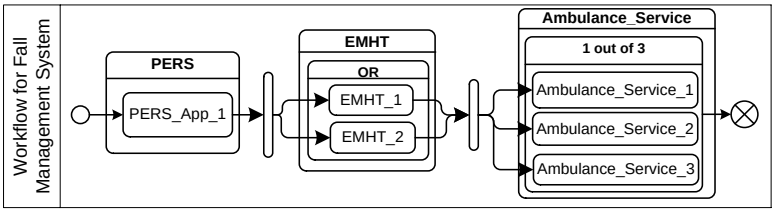

Fig. 4. Example Workflow of the Fall Management System

These skeletons will be combined according to the sequence structures of the asset categories modelled in the workflows. For all of the existing asset instances, certificates containing Metric values stored as evidences must be provided. The E2E Trustworthiness Calculator will extract the metric values from the certificates and use them in the E2E computation. Here, the attribute weights that have been specified by the designer in the first place will be used in order to obtain a single trustworthiness values for the whole system composition.

These relations between the two complementary approaches are illustrated in Fig. 5, which also shows how the generated output supports the evaluation.

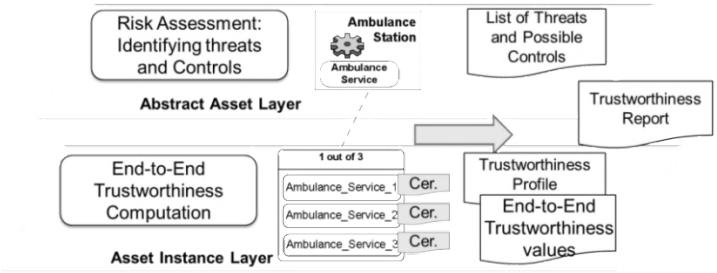

Fig. 5. Example relation between the complementary approaches for trustworthiness evaluation

4 Summary

This paper addresses the problems of the commonly used techniques for evaluating overall trustworthiness in STS. We suggest a combination of two complementary techniques: computational approach and risk management approach. Threats and controls proposed by the risk based approach can be evaluated against actual trustworthiness values, thus substantiating the effectiveness of applied design-time controls. We also presented a tool prototype that supports the assessment of risks related to trustworthiness as well as the evaluation of a system with regard to trustworthiness requirements, aiding the designer in making trustworthiness related decisions and providing evidence and documentation for those decisions.

References

1. Sommerville, I.: Software Engineering, 9th edition, Addison-Wesley, 2011.
2. Lock, R., Sommerville, I.: Modelling and Analysis of Socio-Technical System of Systems, In: 15th IEEE Int'l. Conference on Engineering of Complex Computer Systems, 2010.
3. Dev G. Raheja, Louis J. Gullo: Design for Reliability. Wiley, 2012.
4. Avizienis, A., Laprie, J.-C., Randell, B. and Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing, IEEE Trans. on Dependable and Secure Computing 1 (1), 11-33, 2004.
5. Gol Mohammadi, N., Paulus, S., Bishr, M., Metzger, A., Koennecke, H., Hartenstein, S., Weyer, T. and Pohl K.: Trustworthiness Attributes and Metrics for Engineering Trusted Internet-based Software Systems. In: Cloud Computing and Services Science, (CLOSER Selected Paper), Communications in Computer and Information Science 453. 19-35, 2014.
6. Ali, M., Sabetta, A., Bezzi, M.: A Marketplace for Business Software with Certified Security Properties, In: Cyber Security and Privacy Communications. Computer and Information Science 182. 105-114, 2013.
7. Gol Mohammadi, N., Bandyszak, T., Moffie, M., Chen, X., Weyer, T., Kalogiros, C., Nasser, B., and Mike Surridge: Maintaining Trustworthiness of Socio-Technical Systems at Run-Time, In: Proceedings 11th Int'l. Conference TrustBus, 2014
8. Surridge, M., Nasser, B., Chen, B., Chakravarthy, A., and Melas, P.: Run-Time Risk Management in Adaptive ICT Systems, In: 8th Int'l. Conference on Availability, Reliability and Security (ARES), 102,110, 2013.
9. Christopher, J. A. and Dorofee, A., Managing Information Security Risks: The Octave Approach. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.
10. Fray, I.L.: A comparative study of risk assessment methods, MEHARI & CRAMM with a new formal model of risk assessment (FoMRA) in information systems. In Proc. of the 11th Int'l. Conference on Computer Information Systems and Industrial Management, 2012.
11. ISO/IEC 27005, Information technology — Security techniques — Information security risk management, https://www.iso.org/obp/ui/#iso:std:iso-iec:27005:ed-2:v1:en, 2011
12. OWASP.org (2013), https://www.owasp.org/index.php/Top_10_2013-Top_10 , 2013
13. Hogganvik, I. and Stølen, K.: A graphical approach to risk identification, motivated by empirical investigations, In: Proceedings of the 9th Int'l. Conference on Model Driven Engineering Languages and Systems (MoDELS'06), 2006.
14. Swiderski, F. and Snyder, W.: Threat Modelling. Microsoft Press. 2004.
15. Elshaafi, H., McGibney, J. & Botvich, D. Trustworthiness monitoring and prediction of composite services. Computers and Communications (ISCC), 2012 IEEE Symposium on (pp. 000580–000587), 2012.
16. Zhao, S., Wu, G., Li, Y., and Yu, K.: A Framework for Trustworthy Web Service Management. In: 2nd Int'l. Symposium Electronic Commerce and Security, 479-482, 2009.
17. Malik, Z., and Bouguettaya, A.: RATEWeb: Reputation Assessment for Trust Establishment among Web services. In: VLDB Journal, Vol. 18, Issue 4, pp. 885-911, 2009
18. Cardoso, J., Sheth, A., Miller, J., Arnold, J., and Kochut, K.: Quality of Service for Workflows and Web Service Processes. In: Web Semantics: Science, Services and Agents on the World Wide Web 1 (3), 281-308, 2004.
19. Jaeger, M. C., Rojec-Goldmann, G. and Mühl, G.: QoS Aggregation for Web Service Composition using Workflow Patterns. In: Proceedings of the 8th IEEE Int'l. Enterprise Distributed Object Computing Conf (EDOC 2004), pp. 149 – 159, 2004
20. Hwang, S. Y., Wang, H., Tang, J., Srivastava, J.: A Probabilistic Approach to Modeling and estimating the QoS of web-services-based workflows. In: Journal of Information Sciences 177, 5484–5503, 2007