Information Systems 000 (2017) 1–20

Contents lists available at ScienceDirect

Information Systems

journal homepage: www.elsevier.com/locate/is

Engineering transparency requirements: A modelling and analysis framework

Mahmood Hosseini*, Alimohammad Shahri, Keith Phalp, Raian Ali

Faculty of Science and Technology, Bournemouth University, UK

ARTICLE INFO

Article history:
Received 1 November 2016
Revised 12 December 2017
Accepted 25 December 2017
Available online xxx

Keywords:
Transparency requirements
Transparency management
Requirements engineering

ABSTRACT

Transparency is a requirement that denotes the communication of information that should help audience to take informed decisions. The existing research on transparency in information systems usually focuses on the party who provides transparency and its inter-relation with other requirements such as privacy, security and regulatory requirements. Engineering transparency, however, also requires the analysis of the information receivers' situation and their transparency requirements and the medium used to communicate and present the information. A holistic consideration of transparency will enhance its management and increase its usefulness. In this paper, we provide a novel engineering framework, consisting of a modelling language and nine analytical reasonings, which is meant to represent transparency requirements and detect a set of possible side-effects. Examples of such detections include detecting information overload, information starvation, and transparency leading to biased decisions. We then evaluate the modelling language through a case study and report the results.

1. Introduction

Transparency, as the concept people know and use today, is defined as the extent to which one entity discloses relevant information about its own decision processes, procedures, performance, and functioning [1]. In the domain of requirements engineering, transparency can be defined as a requirement which concerns an information provision or information request amongst the stakeholders of an information system [2], and can be formatted as a user story as follows:

> *"As stakeholder A, I want to get information from stakeholder B, so that I can use the information in my decision making."*

Or as follows:

> *"As stakeholder A, I want to give information to stakeholder B, so that stakeholder B can use the information in their decision making."*

For example, a customer of an insurance company may need to get some information from the company about their cancellation policies, so that the customer can decide whether to take that insurance product from that company. This is an example of transparency for the first user story. In another example, a bank provides information on different current account products and their comparison with each other to the bank customer, so that the bank customer can make an informed decision on what current account product to choose. This is an example of transparency for the second user story [3].

Transparency is meant to provide targeted or public audience with information that helps their decisions about using services and products or dealing with certain social parties. Transparency is by nature a property that includes different stakeholders as it incorporates the communication of information in essence. Such a communication would then need to be carefully designed so that it fits its purpose and avoids properties like redundancy and overload. In this sense, transparency would not only mean making information available but indeed doing that in a way that makes it useful with a secured effort and time from the perspective of both the providers and receivers.

Transparency has been often associated with positive properties such as increasing trust [4] and accountability [5]. Transparency could be seen as a sharing of responsibility mechanism, which means that the communicating parties are collectively accountable when the information related to decision making and the collaborative process is made available. However, despite the benefits, designing correct and efficient mechanisms to implement transparency is more complex than deciding whether to make information available. For example, the space of information could be too large to communicate in a way that can help timely decisions. The choice of the right time and communication style

* Corresponding author.
E-mail address: mhosseini@bournemouth.ac.uk (M. Hosseini).

https://doi.org/10.1016/j.is.2017.12.008
0306-4379/

could also become complex and uncertain. For example, showing lengthy terms and conditions passages to web users is an example of a transparency which causes information overload, rendering the supplied information (and hence, the provided transparency) almost useless to them.

Transparency is often dealt with as being the other side of mainstream requirements such as security and privacy. In addition, it could be also seen as a type of regulatory requirements when law enshrines and enforces it [6]. Such a view of transparency entails that it is somewhat already dealt with when those requirements are engineered. However, this is only partially the case. A holistic engineering of transparency would need to consider additional key pillars of transparency: The audiences and their interest and ability to process the communicated information and how transparency could affect their workflow and decision-making. This means we may have cases where a piece of information passes restrictions made by privacy and security policies and rules but could be expensive and less meaningful if delivered to a particular audience in a particular mode.

The engineering of transparency, therefore, aims to manage it more efficiently and ensure it meets its purpose. Model-driven requirements engineering aims to provide concepts and constructs to project certain concepts and software systems properties, both at the social and technical level [7]. Transparency is one of those socio-technical properties involving the provider(s) of information, the receiver(s), and the communication channel. Mainstream requirements engineering modelling languages might provide a basis for transparency engineering. For example, we can imagine a goal modelling approach which enriches a goal model with additional transparency dimensions, linking transparency to strategic interests and goals and helping actors adapting their strategies according to the fulfilment of their transparency constraints. Similarly, a Business Process Modelling Notation (BPMN) could be augmented with transparency-specific constructs to help a better decision on message exchanges in the workflow that cross-cut different organisational boundaries. However, we will illustrate that due to the complicated nature of transparency and its numerous fine-grained constituents, these augmentations and enrichments fall short of a comprehensive modelling of transparency requirements, and consequently, are deemed inefficient.

This paper builds on our previous work in the domain of transparency engineering. TranspLan, a modelling language for transparency requirements and a demonstration of its usage were initially proposed in [8]. In this paper, we extend it and provide an integrated engineering framework that consists of a consolidated version of the language and provide a set of reasonings. The framework caters for the fact that transparency is a shared property amongst various parties, has unique features in comparison to other classes of requirements and means more than the classic handling of it as a decision of making information available. The analysis part will enable various decision-making processes including the decision on the right level of information to communicate and avoiding the risk of creating bias. We also evaluate the framework through a case study. The foundations of the modelling language and its counterpart analytical kit are built on a review of the literature in multiple domains and presented in [3].

The rest of this paper is structured as follows. Related work in managing transparency is introduced in Section 2. In Section 3, the TranspLan modelling language is introduced along with its constituents, and then several analytic reasonings are proposed on TranspLan modelling language for the analysis of transparency requirements and their possible side-effects. In Section 4, a case study will be utilised for the purpose of evaluating the quality of TranspLan modelling language. Section 5 will be dedicated to the discussions on the evaluation of TranspLan, threats to the validity of this study and possible enhancements and augmentations to the existing TranspLan modelling language. The paper is concluded in Section 6.

2. Related work

Transparency is a long-studied topic in fields of study such as politics, economy, and journalism. In most of these fields of study, transparency of information is considered to be a requirement of citizens [9]. But in the field of requirements engineering, the study of transparency as a requirement is a relatively new topic. While transparency has been mentioned in studies relating to the citing and classification of non-functional requirements, it has seldom been paid a scholarly attention, and has been mostly studied as a second class concept. Furthermore, the existence of two contradicting definitions for transparency in software engineering has complicated the study of transparency as a requirement [10]. Transparency has been used to mean invisibility, e.g., a software system is considered to be transparent when its users do not need to know its underlying mechanisms [11], but it has also been used to mean visibility, e.g., when a software system is considered to be transparent when all functionalities of software are disclosed to users [12].

When transparency is used in its second meaning, sometimes it is argued in two categories of information transparency and process transparency. For example, it is stated that software is transparent if it makes both the information it deals with and the internal functioning process transparent, called information transparency and process transparency respectively [13,14]. However, since being transparent about processes means giving information about those processes, one can still consider process transparency to be a subcategory of information transparency. Therefore, we use the expression “information transparency” in this paper to refer to all transparency types where information is being disclosed.

From the perspective of requirements engineering, transparency is commonly categorised as a non-functional requirement (NFR), because it is seen to be orthogonal to the software functionality and is considered as a quality issue, and because software is generally seen to be functional without a special consideration of transparency [14]. Considering transparency as an NFR, it is argued that it can subjectively be satisficed [15]. Furthermore, as an NFR, transparency is aided by other non-functional requirements such as accessibility, usability, informativeness, understandability, and auditability [14]. That being said, it should be noted that transparency requirements can relate to both functional and non-functional requirements within a software system. For example, notifying a software system user that their feedback has been considered in the new release of the product is a functional transparency requirement, while revealing to them the criteria leading to making this decision is a non-functional transparency requirement, i.e., an informative transparency.

Some works on transparency requirements have been conducted by the researchers in the field of requirements engineering. For example, using the NFR Framework, a software transparency softgoal interdependency graph has been proposed which illustrates the interdependencies between transparency requirements and other NFRs [16]. Similarly, it is argued in [17] that transparency requirements can be managed using the NFR Framework [16] and i^* modelling [18]. However, they also admit that i^* is not the final answer to transparency, as there are shortcomings to be addressed.

Another study on transparency requirements argues that organisations must know what transparency is and how they can demonstrate transparency [13]. For this purpose, a transparency ladder is presented, which contains the following five NFRs of accessibility, usability, informativeness, understandability, and auditability, and it is argued that these five NFRs must be achieved in order to reach transparency. By using Github as an example

Please cite this article as: M. Hosseini et al., Engineering transparency requirements: A modelling and analysis framework, Information Systems (2017), https://doi.org/10.1016/j.is.2017.12.008