

Engineering Transparency Requirements: a Modelling and Analysis Framework

Mahmood Hosseini, Alimohammad Shahri, Keith Phalp, Raian Ali

Faculty of Science and Technology, Bournemouth University

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.

Keywords: Transparency Requirements, Transparency Management, Requirements Engineering

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
5 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:

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

¹Corresponding Author: Mahmood Hosseini. mhosseini@bournemouth.ac.uk

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 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
60 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 frame-
65 work 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
70 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
75 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
80 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 jour-
nalism. In most of these fields of study, transparency of information is considered to be a re-
quirement of citizens [9]. But in the field of requirements engineering, the study of transparency
85 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 ex-
istence 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,
90 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
95 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 pro-
cess transparent, called information transparency and process transparency respectively [13, 14].
However, since being transparent about processes means giving information about those pro-
cesses, one can still consider process transparency to be a subcategory of information trans-
100 parency. 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 function-

ality and is considered as a quality issue, and because software is generally seen to be functional
105 without a special consideration of transparency [14]. Considering transparency as an NFR, it is
argued that it can subjectively be satisfied [15]. Furthermore, as an NFR, transparency is aided
by other non-functional requirements such as accessibility, usability, informativeness, under-
standability, and auditability [14]. That being said, it should be noted that transparency require-
ments can relate to both functional and non-functional requirements within a software system.
110 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
115 field of requirements engineering. For example, using the NFR Framework, a software trans-
parency softgoal interdependency graph has been proposed which illustrates the interdependen-
cies 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^* mod-
elling [18]. However, they also admit that i^* is not the final answer to transparency, as there are
120 shortcomings to be addressed.

Another study on transparency requirements argues that organisations must know what trans-
parency is and how they can demonstrate transparency [13]. For this purpose, a transparency lad-
der is presented, which contains the following five NFRs of accessibility, usability, informativ-
ness, understandability, and auditability, and it is argued that these five NFRs must be achieved
125 in order to reach transparency. By using Github as an example of a transparent environment,
transparency is shown to have the ability to reveal users' needs and requirements [19, 20].

In another study [21], and for eliciting transparency requirements, a novel approach is used to
capture transparency requirements of stakeholders through an Argumentation Framework [22].
It is also advocated that to provide transparency, it must be dealt with in the context of require-
ments specification [14]. Another study illustrates the difficulties of presenting the transparency
130 catalogue and provide solutions for them [23], while it is illustrated [17] that the evaluation of
transparency has a close relationship with "Quality Questions" known as 5WH [24].

Before identifying the need for a new modelling language for modelling and analysing trans-
parency requirements in an information system, we attempted to augment or modify existing
135 modelling languages in order to make them suitable for such modelling and analysis. As dis-
cussed earlier, i^* modelling, which was also proposed to have the base requirements of trans-
parency modelling [17], was considered in particular.

We made several attempts to augment this modelling language with the constituents of trans-
parency so that it could also be used for modelling transparency. These attempts, however, failed
140 for several reasons. First, i^* is goal-oriented, while transparency is information-oriented. That
is to say, while i^* modelling focuses mainly on goals and how different tasks can help achieve
those goals, transparency focuses on stakeholders and how information is exchanged amongst
them. Second, transparency itself is considered a softgoal in i^* , meaning that it is part of a bigger
picture in which tasks are conducted to reach other goals of the actors involved. On the other
145 hand, transparency itself becomes the main focus of any information exchange and other possible
goals of actors become inconsequential in this manner. Third, i^* treats information as a resource
which circulates amongst different actors, and in i^* , resource modelling has no elaboration and
therefore no significance. In transparency, however, information plays a vital role and is a central
entity. This is because transparency is an information-oriented property, so an extension of i^* for
150 transparency requirements will not be a mere extension, given the proportion needed for accom-

modating extra concepts. Finally, several fine-grained attributes of transparency with regards to meaningfulness, usefulness, and quality of information could not be modelled using i^* .

Similarly, the extensions of i^* (such as TROPOS) or BPMN may not be suitable for the task at hand, because such extensions are also requirements-driven or activity-driven, and not information-driven. These two aspects can be integrated but we would not see one as an extension of the other. This explains our choice of thinking of transparency independently at this stage and proposing a modelling language expressly tailored to its characteristics without excluding the possibility of integrating it with mainstream requirements languages at some stage.

3. Transparency Requirements Modelling in TranspLan

We have designed TranspLan in order to help information system analysts in capturing and analysing transparency requirements. The design of TranspLan is based on our four proposed transparency reference models [3]. These reference models are in turn based on an extensive literature study on transparency in different fields of study such as politics, economy, journalism, etc. These four reference models are:

1. Transparency Actors Wheel: which illustrates transparency stakeholders and the information flow amongst them
2. Transparency Depth Pyramid: which illustrates the depth and meaningfulness of information
3. Transparency Achievement Spectrum: which illustrates the steps necessary to achieve a useful transparency
4. Transparency Information Quality: which illustrates the information quality dimensions for transparency provision

TranspLan consists of Stakeholders' Information Exchange Layout Diagram (Shield diagram) for the visual representation of information exchanges amongst stakeholders and their transparency requirements. TranspLan is also accompanied by two descriptive specification models for information elements and stakeholders, called INFORMATION eLEMENT Transparency Specification (Infolet specification) and Stakeholders' Information Transparency REquirements Specification (Sitreq specification), respectively. These specification models explain the information elements and the stakeholders with their elicited transparency requirements in the Shield diagram.

3.1. Modelling Constituents and Representations

The TranspLan language is mainly built based on three different constituents: *stakeholders*, *information elements*, and the *relationships* between stakeholders and information elements. Relationships can be decomposed using *decomposition relations*. An *information exchange* is a combination of all these constituents and illustrates the flow of information amongst different stakeholders. These constituents are described as follows.

- **Stakeholders** are the people, departments, organisations, etc., which are involved in providing, receiving, or requesting transparency in any information exchange amongst stakeholders. When categorising stakeholders, they are commonly represented as one entity, e.g., Student or Finance Department. However, the exchanged information within an information exchange system may concern all the stakeholders within that system (referred to as *All* in TranspLan), and it may also concern the public audience (referred to as *Public* in TranspLan).

- 195 • **Information elements** are pieces of information exchanged amongst stakeholders. Stakeholders' transparency requirements affect the way information elements should be formed and presented to other stakeholders. Information elements have a type, which is related to their transparency meaningfulness. These types can be the *data* type, the *process* type, or the *policy* type. Examples of information elements are privacy policy statements, mortgage documents, and application forms.
- 200 • **Stakeholder-information relationships** exist between stakeholders and information elements, and they describe how the information element is associated with the stakeholder. The *production* relationship denotes that the stakeholder produces the information element for other stakeholders. The *obligation* relationship denotes that the stakeholder provides the information element based on coercive supply or requests the information element based on legal demands. The *optionality* relationship denotes that the stakeholder provides the information element based on voluntary supply or requests the information element based on personal demands. The *restriction* relationship denotes that the information element should not be available to the stakeholder. The *undecidedness* relationship denotes that the relationship between the stakeholder and the information element is not known or decided yet. For example, a bank (i.e., the stakeholder) must provide a mortgage guide (i.e., information element) to their customer (i.e., the other stakeholder). Therefore, there will be two relationships, one provision relationship between the bank and the mortgage document and one obligation document between the mortgage document and the bank customer.
- 210 • **Decomposition relations** exist between some relationships and can be one of the following: the *and* decomposition relation, the *or* decomposition relation, and the *xor* (exclusive or) decomposition relation.
- 215 • **Information exchanges** illustrate the flow of information from an information provider to an information receiver or requester. For example, the bank, their customer, and the mortgage document together constitute an information exchange between these two stakeholders. An information exchange system is a collection of all information exchanges in an information system.
- 220

3.2. *TranspLan* Mathematical Definition

225 The *TranspLan* language and its constituents can be defined using the ordinary mathematical language as follows:

Definition 1 (Information element). Let $IE = \{ie_1, ie_2, \dots, ie_m\}$ be the set of information elements, and IE_Label and IE_Name be sets of unique labels and names respectively. Every $ie_i \in IE$ can be defined as follows:

$$230 \quad IE = \{ie \mid ie = (ietype, ielabel, iename, ieused) \wedge ietype \in IE_type \wedge ielabel \in IE_Label \wedge iename \in IE_name \wedge ieused \subset ielabel\}$$

$$IE_type = \{data, process, policy\}$$

Definition 2 (Stakeholder). Let $S = \{s_1, s_2, \dots, s_n\}$ be the set of stakeholders, $IE = \{ie_1, ie_2, \dots, ie_m\}$ be the set of information elements, and $R = \{r_1, r_2, \dots, r_l\}$ be the set of stakeholder-information relationships. The set of stakeholders and two subsets of S , called PS and RS , can be defined as follows:

$$\begin{aligned}
S &= \{s \mid s \text{ is a stakeholder}\} \\
PS &= \{s \mid s \in S \wedge ie \in IE \wedge (s, ie, \text{production}) \in R\} \\
RS &= \{s \mid s \in S \wedge ie \in IE \wedge rt \in \{\text{obligatory, optional, restricted, undecided}\} \wedge (s, ie, rt) \in R\}
\end{aligned}$$

240 **Definition 3 (Stakeholder-information relationship).** Let $R = \{r_1, r_2, \dots, r_l\}$ be the set of relationships where each relationship is between stakeholder $s_i \in S$ and information element $ie_j \in IE$. Every $r_i \in R$ can be defined as follows:

$$\begin{aligned}
R &= \{r \mid r = (s, ie, rtype) \wedge s \in S \wedge ie \in IE \wedge rtype \in R_type\} \\
R_type &= \{\text{production, obligatory, optional, restricted, undecided}\}
\end{aligned}$$

245 **Definition 4 (Decomposition relation).** Let $Rel = \{rel_1, rel_2, \dots, rel_k\}$ be the set of relations where each relation is between two or more relationships $R_1, R_2, \dots, R_j \in R$. Every $Rel_i \in Rel$ can be defined as follows:

$$\begin{aligned}
Rel &= \{rel \mid rel = (r_1, r_2, \dots, r_j, reltype) \wedge r_1, r_2, \dots, r_j \in R \wedge reltype \in Rel_type\} \\
Rel_type &= \{\text{and, or, xor}\}
\end{aligned}$$

250 **Definition 5 (Information exchange).** Let $IEX = \{iex_1, iex_2, \dots, iex_i\}$ be the set of information exchanges amongst stakeholders where one stakeholder $s \in PS$ produces some information elements $IESet \subset IE$ that is received or requested by a group of other stakeholders $RSSet \subset RS$ and $s \notin RSSet$. Every information exchange iex_i can be defined as follows:

255
$$IEX = \{iex \mid iex = ((s_i, ie_i, r_i), (s_j, ie_i, r_j)) \wedge s_i \in PS \wedge s_j \in RS \wedge r_i = \text{production} \wedge (s_i, ie_i, r_i), (s_j, ie_i, r_j) \in R\}$$

3.3. Shield Diagram

The Shield diagram is the graphical representation of the TranspLan language. The constituents of the TranspLan language can be illustrated in the Shield diagram as follows.

3.3.1. Stakeholders

260 Stakeholders are captured in one of the four following ways. It can be one stakeholder, all stakeholders within an information exchange system (for the purpose of facilitating a more efficient, clutter-free visual design) possibly except a few (for referring to those stakeholders who are excluded from the information exchange), and public stakeholders (for referring to the public, i.e., all stakeholders inside and outside the information exchange system under study)
265 (See Figure 1).

3.3.2. Information elements

Information elements capture the type of information element (i.e., data, process, and policy), a unique information element label, its name, and a list of all the other information element tags which use, partly or completely, the current information element, which can be used to track how
270 the information travels and how these information elements are related to each other (See Figure 1).

Stakeholder's Information Transparency REquirements Specification (Sitreq)						
Stakeholder's Name: <i>Stakeholder's Name</i>						
IE Label	IE Name	Relationship	Requirement Description	Transparency Requirement Type	Transparency Meaningfulness Type	Stakeholders Involved
<i>IE label</i>	<i>IE name</i>	<i>Relationship type</i>	<i>A brief description of the stakeholder's requirements regarding the IE</i>	<i>The transparency requirement of the stakeholder regarding the IE</i>	<i>The meaningfulness type requirement regarding the IE</i>	<i>List of stakeholders in this information exchange</i>

Table 1: Stakeholder's Information Transparency REquirements Specification (Sitreq)

3.3.3. Stakeholder-information relationships

Stakeholder-information relationships capture whether the stakeholder is producing the information, or receiving the information on an obligatory or optional basis. They also capture currently unknown relationships and restricted relationships. Arrows are intentionally chosen to be dotted in order to emphasise that such information flow may or may not serve its transparency purpose because its usefulness must be decided through complicated procedures and involvement with stakeholders which simply cannot be captured through such diagrams (See Figure 1).

3.3.4. Decomposition relations

Decomposition relations describe the relationship amongst relationships and can be *and* (the default relation), *or*, and *exclusive_or* (See Figure 1).

3.3.5. Information exchange system

An information exchange system captures the following four parts: its name, its description and extra notes, a list of all stakeholders in the information exchange system including two pre-defined *All* and *Public* stakeholders, and all the information exchanges amongst the stakeholders (See Figure 1)..

3.4. Sitreq Specification

Every stakeholder in the Shield diagram is accompanied by Stakeholder's Information Transparency REquirements Specification (Sitreq), as illustrated in Table 1. Sitreq is a descriptive tool for stakeholders and their transparency requirements in the Shield diagram. Sitreq explains how stakeholders are related to certain information elements, their transparency requirements on those information elements, and other stakeholders involved in the process.

3.5. Infolet Specification

Every information element in the Shield diagram is accompanied by a INFOrmation eLEMENT Transparency Specification (Infolet), as illustrated in Table 2. Infolet is a descriptive tool for information exchanges in the Shield diagram. It describes each information element (IE) in the diagram, providing more in-depth information on them. Infolet is meant to capture all the four reference models of transparency, along with general modelling information required for each information element, as follows. The numbers on parentheses illustrate the corresponding segments in Infolet.

1. General modelling requirements (1, 2, 4, 5, 13)


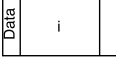



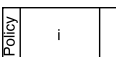

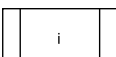

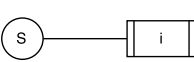

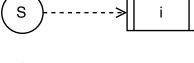


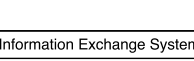

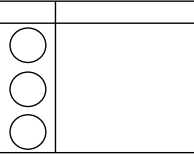


Visual Syntax	Semantics	Visual Syntax	Semantics
Stakeholders		Information Elements	
	stakeholder <i>S</i>		information element <i>i</i> containing data
	all stakeholders in the diagram		information element <i>i</i> containing a process
	all stakeholders in the diagram (except <i>S1</i> , <i>S2</i>)		information element <i>i</i> containing a goal
	public stakeholders		unspecified information element <i>i</i>
Relationships		Stakeholder/Information Relations	
	AND relationship		stakeholder <i>S</i> producing information <i>i</i>
	OR relationship		stakeholder <i>S</i> requesting information <i>i</i>
	eXclusive OR relationship		stakeholder <i>S</i> receiving information <i>i</i>
			information <i>i</i> not intended for stakeholder <i>S</i>
Relation Types		Information Exchange System	
	<i>Obligatory</i> relationship: Coercive Provision / Legal Demand		information exchange system boundary with the list of stakeholders
	<i>Optional</i> relationship: Voluntary Provision / Personal Demand		
	<i>Undecided</i> relationship: Unknown Provision or Demand		

Figure 1: Building blocks of *Shield* and their interpretations

2. Transparency Depth Pyramid (3)
3. Transparency Actors Wheel (6, 7, 8, 9, 10)
4. Transparency Information Quality (11)
5. Transparency Achievement Spectrum (12)

305

3.6. Transparency Requirements Analysis Using *TranspLan*

The modelling language, *TranspLan*, and its components, the *Shield* diagram and *Sitreq* and *Infolet* specifications, provide a viable solution for addressing several problems that an information system may encounter during transparency provision, because they enable automated transparency analysis and tool support. The automated analysis enables algorithmic investigation of transparency in order to identify issues such as transparency shortage or abundance in an information exchange system and amongst stakeholders [25, 2].

310

INFORMATION eLEMENT TRANSPARENCY SPECIFICATION (Infolet)			
<i>Information Element Label, Name, Description, Inter-relations and Type (Transparency Meaningfulness)</i>	① Information Element Label A unique label for IE identification	② Information Element Name A name selected for the IE	③ Information Element Type IE type can be selected from: {Data, Process, Policy}
	④ Information Element Description A brief description of IE and its content		
	⑤ List of Other Information Elements Using This Information Element A list of IE labels and names which use part of all of the current IE		
<i>Information Element Stakeholders Classification and Their Information Provision/Request/Restriction Type (Transparency Stakeholders)</i>	⑥ Information Element Creator/Authority Information entity responsible for creating, producing, and rendering IE		
	⑦ Information Element Provider Stakeholder who provides the information is listed.		
	⑧ List of Stakeholders Receiving Information Element and Information Element Provision Type Stakeholders who receive the information are listed IE provision type can be selected from: {Coercive, Voluntary}		
	⑨ List of Stakeholders Requesting Information Element and Information Element Request Type Stakeholders who request the information are listed IE request type can be selected from: {Legal, Personal}		
	⑩ List of Stakeholders with Restricted Access to Information Element and Restriction Type Stakeholders who cannot access the information are listed Restriction type can be selected from: {Secrecy, Privacy, Anonymity, Other}		
<i>Information Element Quality Control (Transparency Quality)</i>	⑪ Information Element Quality Control (Sound, Dependable, Useful, Usable) <input type="radio"/> Free of Error <input type="radio"/> Concise Rep. <input type="radio"/> Completeness <input type="radio"/> Consistent Rep. <input type="radio"/> Timeliness <input type="radio"/> Security <input type="radio"/> App. Amount <input type="radio"/> Relevancy <input type="radio"/> Understandability <input type="radio"/> Objectivity <input type="radio"/> Interpretability <input type="radio"/> Accessibility <input type="radio"/> Believability <input type="radio"/> Ease of Manipulation <input type="radio"/> Reputation <input type="radio"/> Value-added IE qualities are checked when these qualities are met.		
<i>Information Element Level of Achievement (Transparency Usefulness)</i>	⑫ Information Element Level of Achievement <input type="radio"/> Information Availability (Information is made available to the stakeholders) <input type="radio"/> Information Interpretation (Information is appropriately interpreted for the stakeholders) <input type="radio"/> Information Accessibility (Information is easily accessible by the stakeholders) <input type="radio"/> Information Perception (Information is perceived credible by the stakeholders) <input type="radio"/> Information Understandability (Information is comprehended by the stakeholders) <input type="radio"/> Information Acceptance (Information is believed and accepted by the stakeholders) <input type="radio"/> Information Actionability (Information helps stakeholders in their informed decision-making) IE level of achievement is checked.		
<i>Information Element Additional Data</i>	⑬ Information Element Notes Further notes about IE, samples, links, etc.		

Table 2: INFORMATION eLEMENT TRANSPARENCY SPECIFICATION (Infolet)

We acknowledge the cognitive and social aspects of the side-effects of transparency, such as bias and information asymmetry, require a far deeper psychological analysis and are therefore difficult to enact. As a result, in these analyses we do not intend or claim to detect all instances of the property being detected but rather factual and business-related cases. Furthermore, the analyses can yield results which may need further qualification from the analyst to consider factors which are not in the model, including the nature of the audience and the additional information and background they could be already having. This would be a common expectation of analyses dealing with human factors. It is also worth mentioning that it is possible to have interaction effects amongst transparency properties where one would think of considering various confounding and moderating factors and the need to consider these transparency property checks as just predictors and recommendations to further diagnose the issue rather than giving a definite answer.

In the following subsections, nine analytical reasonings are provided for the analysis of transparency requirements. For each reasoning, we specify the problem definition and purpose, the benefits of the reasoning for providing better transparency to stakeholders, an example of a situation where the problem related to the reasoning may occur, and a list of transparency properties that can be checked for the discovery of the problem. A summary of the analytical reasonings will also be provided at the end of this subsection.

3.6.1. *Transparency Meaningfulness Mismatch*

Problem Definition and Purpose: Transparency meaningfulness mismatch happens when the level of meaningfulness provided by a stakeholder does not match with the level that is requested by another stakeholder. Failure in reaching the required transparency level (e.g., disclosing the actions without giving the rationale behind them) may reduce accountability, while exceeding the required transparency level (e.g., disclosing the reasons for a particular action when only the data obtained from the action is needed) may introduce various workarounds in the information system [26].

Benefit: Eliminating transparency meaningfulness mismatch means that stakeholders get the information depth they require which in turn can minimise information overload.

Example: An employee is asked through an email to send their bank account details to the finance department of their organisation (i.e., they are provided with “data”). However, the employee needs to know why this information has been requested (i.e., they need “policy”). Another employee has been asked to complete an online form through an email and complete instructions on how to fill in the form has been provided (i.e., they are provided with “process”). However, the employee has already filled in the same online form several times and need not be reminded every time of the process (i.e., they only need “data”). Both cases are examples of transparency mismatch.

Transparency Properties to Check: In order to check transparency meaningfulness mismatch, the information provider’s provided information depth and information receiver’s required information depth must be checked against one another. These information properties can be found in Sitreq documents.

3.6.2. *Transparency Leakage*

Problem Definition and Purpose: Transparency leakage refers to the availability of information elements to stakeholders who initially were not meant to receive that information because of the restricted nature of other stakeholders’ transparency requirements. Transparency leakage

can produce several adverse effects, e.g., it can affect stakeholders' trust in the information system negatively and it can influence trading behaviour and market efficiency in financial systems [27].

360 **Benefit:** Eliminating transparency leakage can increase trust in stakeholders. It can also decrease privacy and security concerns related to the disclosed information.

Example: A human resources report includes a list of employees who have been working fewer hours than they should. The report is meant for high level managers, and not for the employees of the organisation. In particular, it must not be viewed by the employees whose names appear in the list. However, if the report is accidentally put in the public domain and 365 allows employees to access it, it leads to transparency leakage.

Transparency Properties to Check: In order to check transparency leakage, first restricted access to information elements must be checked and a list of stakeholders with restricted access to these information elements must be created. Then their access to other information elements 370 which contain a part of or all the information in that particular information element must be investigated, and such occurrences must be listed as transparency leakage instances. These information properties can be found in Infolet documents and can also be deduced from the Shield diagram.

3.6.3. Information Overload (Infobesity or Infoxication)

375 **Problem Definition and Purpose:** Information overload, sometimes referred to as infobesity [28] or infoxication [29], happens when more information is given to a consumer than they need. Investigating the four reference models for transparency, it can be observed that Transparency Depth Pyramid can capture information overload. Infobesity can happen both horizontally across the model, i.e., disclosing too much data than one needs or disclosing too much process than one 380 needs. It can also happen vertically along the model, i.e., disclosing processes and policies when one only needs data, or disclosing policies when one only needs data and processes. The latter case also leads to transparency meaningfulness mismatch, which was discussed in Subsection 3.6.1.

Benefit: Minimising information overload helps reduce the cognitive overload of the stakeholders and can help them in their decision making processes. 385

Example: An employee of an organisation receives several work-related emails per hour, and they have to answer them all. However, they cannot cope with the amount of information they get, and sometimes miss certain emails as a result. The employee may have to develop certain coping strategies, e.g., prioritising, multitasking, satisficing, refusing, queuing, and delegating 390 [30, 31], but the main problem, i.e., information overload, remains unsolved.

Transparency Properties to Check: In order to check information overload, first the provided and required information depths must be checked against each other, and if provided information depth exceeds the required information depth, then it can be concluded that information overload has happened. These information properties can be found in Sitreq documents. 395 Second, *information actionability* and *information appropriate amount* must be checked, and if they are both not achieved, then it can be concluded that information overload might happen. These information properties can be found in Infolet documents.

3.6.4. Information Starvation (Infoforexia)

400 **Problem Definition and Purpose:** Information starvation, also referred to as infoforexia (in contrast with infobesity), happens when less information is given to a consumer than they need. Amongst other adverse effects, it has been shown to be the source of several problems in the

workplace [32]. Similar to infobesity, it can be observed that Transparency Depth Pyramid can capture information starvation. Inforexia can also happen both horizontally across the model, i.e., disclosing too little data than one needs or too little policy than one needs. It can also happen
405 vertically along the model, i.e., disclosing only data when one needs processes and policies, or disclosing processes when one needs policies. The latter case also leads to transparency meaningfulness mismatch, which was discussed in Subsection 3.6.1.

Benefit: Minimising information overload helps stakeholders in their decision making processes by providing them with the appropriate amount of information which is complete for the
410 task at hand.

Example: A customer is in the process of buying a car. They need certain information before they can decide whether to buy a particular car (e.g., the price, the number and amount of instalments, the final calculated price, possible mileage restrictions). If they do not have access to all the information they need and some of their questions remain unanswered, then information
415 starvation will happen, which will impair their decision making.

Transparency Properties to Check: In order to check information starvation, first the provided and required information depths must be checked against each other, and if provided information depth falls below the required information depth, then it can be concluded that information starvation has happened. These information properties can be found in Sitreq documents.
420 Second, *information actionability* and *information completeness* must be checked, and if they are both not achieved, then it can be concluded that information starvation might happen. These information properties can be found in Infolet documents.

3.6.5. Detecting Bias

Problem Definition and Purpose: Bias is the process in which a person seeks information
425 to confirm a preconceived belief. Bias happens because human beings have a tendency to avoid information which can disprove their already held beliefs and throw their decisions into question. In other words, bias allows people to see the world around them the way they want to see it.

Bias normally is the result of one of the three preformed sources [33]:

- Bias source 1: Information processing shortcuts, in which people make educated guesses
430 instead of fully considering the information.
- Bias source 2: Social influences or beliefs, which assume that something is true because their belief system tells them so.
- Bias source 3: Motivational factors, which lead people to search for information which supports their current ideas.

The first source, therefore, is about holding, or having access to, incomplete information,
435 while the second and third sources are mainly about existing beliefs which can affect new information acceptability. Their difference, however, is in the deliberateness of the bias, because while the second type of bias source is unintentional and subconscious, the third one is intentional and self-serving. Looking at the four reference models of transparency, it can be observed
440 that bias has to do both with information quality and with information acceptance, which is a step in transparency achievement.

Please note that in the second and third type of bias source, the information does not reach the actionable stage because it is not accepted by the information receiver. Furthermore, this analytical reasoning does not capture all instances of bias, as it requires a deep psychological

445 and socio-contextual analysis before a firm conclusion can be drawn. It only acts based on the
provided transparency requirements analysis of the information flow.

Benefit: Minimising bias in information flows has several advantages. For example, capturing the first bias source will help stakeholders to look for alternative sources of information where they can have access to more comprehensive information and can therefore make better
450 decisions. Capturing the second bias source will allow information providers to break information receivers free of their inadequate social influences and beliefs, which also leads to a more vigorous decision making process. Capturing the third bias source allows information systems to propagate information in a more efficient way where stakeholders will have access to all sorts of approval and disapproval information at the same time, and can decide for themselves which
455 one to accept.

Example: A social platform which disseminates news to its users can estimate a particular user's bias in a given subject by looking through the news pieces they have clicked on and the time they have spent reading those news pieces. The social platform can then expose this user to news pieces from other sources with alternative stories and viewpoints so that the user can also
460 have access and probably read these news pieces and possibly change their biased viewpoint.

Transparency Properties to Check: In order to check bias, *information free-of-error*, *information completeness*, or *information objectivity* must be checked, and if one of them is not achieved, then it can be concluded that bias type one might happen. Furthermore, if *information understandability* is achieved but *information acceptance* is not achieved, and *information believability* is also not achieved, then it can be concluded that bias type two or type three might
465 happen. These information properties can be found in Infolet documents.

3.6.6. Information Asymmetry

Problem Definition and Purpose: Information asymmetry refers to the condition in which one party has access to information while another party does not. One way to cause information
470 asymmetry is for information providers to restrain information (or part of it) from information receivers. Information asymmetry can also happen when different information receivers have different amounts of information received by them. Information asymmetry has been shown to have adverse effect on information consumers [34, 35].

Benefit: Minimising information asymmetry helps stakeholders to decide based on similar information and eliminates information hegemony of a group of stakeholders over others.
475

Example: An insurance company may talk hours about the benefits of a new service, while deliberately refraining from discussing its long-term costs or limitations. Also, when different people watch different news networks or read different newspapers, the information they obtain will vary from one another, which will lead to information asymmetry. Both are examples of
480 information asymmetry.

Transparency Properties to Check: In order to check information asymmetry, *information completeness* must be checked, and if this is not achieved, then it can be concluded that information asymmetry might happen. Furthermore, if different media is used by different stakeholders to obtain the same information, then it can be concluded that information asymmetry might happen.
485 These information properties can be found in Infolet and Sitreq documents respectively.

3.6.7. Unidirectional and Bidirectional Transparency

Problem Definition and Purpose: Unidirectional transparency, also called static transparency, occurs when information flows exists only from the information provider to the information receiver, with no information flow back to the information provider from the infor-

490 mation receiver in the form of comments, corrections, suggestions, etc. [36, 37]. Bidirectional
transparency, also called dynamic transparency, occurs when information flow exists in both di-
rections from and to the information provider and information receiver [36, 37]. As discussed
earlier, unidirectional transparency, e.g., in the form of computer-mediated transparency, has the
potential to threaten trust [38]. TranspLan can spot unidirectional transparency and therefore,
495 makes software engineers dealing with transparency requirements aware of its existence.

Benefit: Achieving bidirectional transparency is beneficial for stakeholders as it can increase
trust between them. It also keeps the information exchange channel open between stakeholders
in which they can keep each other updated with relevant information.

Example: A university lecturer marks their students' assignments and provides comments
500 on their assignments to them. However, he does not get any feedback on their marking and com-
ments from the students. Consequently, the lecturer might think that everything is satisfactory
and he does not deem it as necessary to modify their marking process or the depth and breadth
of their comments.

Transparency Properties to Check: In order to check unidirectional transparency, the ex-
505 istence of information exchange between each two stakeholders must be checked, and if there
is no information exchange between them, then it can be concluded that there is unidirectional
transparency. These information properties can be found in Infolet documents and can also be
deducted from the Shield diagram.

3.6.8. Social, Target, and Organisational Transparency

510 **Problem Definition and Purpose:** The literature on transparency discusses three types of
transparency, as follows [14]:

- Social transparency, which aims at the general public
- Target transparency, which aims at the consumers of a certain service or product
- Organisational transparency, which aims at an organisation's stakeholders

515 TranspLan can capture the first two types of transparency, based on the structure of its infor-
mation exchanges. Organisational transparency, on the other hand, can only be captured through
TranspLan if an information exchange system is modelled for each and every one of its stake-
holders, but there is no analytical reasoning which can capture that based on TranspLan.

Benefit: Social transparency guarantees that the information is publicly available and this
520 ensures that every stakeholder interested in obtaining and analysing information can have proper
access to that. Target transparency makes the information available to information consumers
who probably need that information, and therefore guarantees that their information requirement
is met.

Example: A charity organisation publishes their annual tax review and their total sales in-
525 come on the website of the organisation for the general public in order to achieve social trans-
parency. The same organisation also provides a portal for their gift aid donors and regular reg-
istered donors to track how their monetary and commodity donations have been spent on the
cause of the charity, therefore attempting to achieve target transparency. Assuming that the char-
ity organisation puts enough information for all its stakeholders on its website, organisational
530 transparency is also achieved.

Transparency Properties to Check: In order to check social transparency, the existence of
information provision to the public audience must be checked. Other instances of information

provision should be considered as target transparency because they have their specific audience. Such information provision can be found in Infolet documents and can also be deduced from the Shield diagram.

3.6.9. Opaque and Clear Transparency

Problem Definition and Purpose: In another categorisation of transparency, two faces of transparency are discussed [39]. One is opaque transparency, which is providing information which is not clearly understandable, and the other is clear transparency, its opposite. TranspLan can capture these two types of transparency as well, thanks to its fine-grained information quality dimensions and Transparency Achievement Spectrum.

Benefit: Capturing opaque transparency can help information providers to amend and modify their information so that it meets its transparency requirements.

Example: A tour agency provides a travel brochure for its customers, which specifies travel destinations, accommodation opportunities, prices, and so on. However, the travel brochure is not clear in terms of its service fees and commissions which may vary based on the number of passengers, the chosen destination and the selected travel type. Such opaque transparency may actually hinder potential customers from further investigation and from choosing the mentioned travel agency as their service provider.

Transparency Properties to Check: In order to check opaque transparency, *information understandability* must be checked and it should also be checked whether the information has passed this stage, and if this is not achieved, then it can be concluded that opaque transparency exists on the provided information. These information properties can be found in Infolet documents.

3.6.10. A Summary of the Transparency Requirements Analysis

A summary of the aforementioned analytical reasonings for transparency requirements using TranspLan is provided in Table 3. The table provides each reasoning with the relevant information source or sources in TranspLan, its detection condition, and the resulting action. For the resulting action, an error should be produced when there is a certainty that the reasoning has identified an issue with transparency provision; a warning should be produced when there is a possibility that the reasoning has detected an issue with transparency provision; and a notification should be produced when a transparency type has been perceived.

4. Evaluating the Quality of TranspLan

In this section, TranspLan will be evaluated in order to see if quality models can be created using TranspLan for transparency requirements. Several individual models will be made using TranspLan, and each individual model is evaluated according to criteria inspired by the SEQUAL framework [40]. The aim of this evaluation is to draw conclusions whether it is possible to construct models that meet these criteria using TranspLan.

A case study involving asking software engineers to do modelling and scenario building is used. This case study concentrates on the quality and fitness of TranspLan in modelling transparency requirements. The main aim of this evaluation is to investigate the perspective of the software engineers in terms of the quality of the transparency modelling language in capturing, modelling, and analysis transparency requirements as well as its quality for the management of

Table 3: Summarising Analytical Reasonings on TranspLan

Reasoning	Relevant Information Source(s)	Detection Condition(s)	Resulting Action
Transparency Meaningfulness Mismatch	Sitreq	The information provider's <i>provided information</i> depth and information receiver's <i>required information</i> depth must be checked	Produce an Error
Transparency Leakage	Infolet	<i>Restricted access</i> to information elements must be checked against stakeholders who have access to them	Produce an Error
Information Overload (Infobesity or Infotoxication)	Sitreq and Infolet	1) The <i>provided and required information</i> depths must be checked against each other 2) <i>Information actionability</i> and <i>information appropriate amount</i> must be checked	Produce an Error or Warning
Information Starvation (Inforexia)	Sitreq and Infolet	1) The <i>provided and required information</i> depths must be checked against each other 2) <i>Information actionability</i> and <i>information completeness</i> must be checked	Produce an Error or Warning
Detecting Bias	Infolet	1) <i>Information free-of-error, information completeness, or information objectivity</i> must be checked 2) <i>Information understandability</i> and <i>information acceptance</i> must be checked along with <i>information believability</i>	Produce a Warning
Information Asymmetry	Infolet and Sitreq	1) <i>Information completeness</i> must be checked 2) The <i>media used</i> by different stakeholders to obtain the same information must be checked	Produce a Warning
Unidirectional and Bidirectional Transparency	Infolet	The existence of <i>information exchange between each two stakeholders</i> must be checked	Produce a Warning or Notification
Social, Target, and Organisational Transparency	Infolet	The existence of <i>information provision to the public audience</i> must be checked	Produce a Notification
Opaque and Clear Transparency	Infolet	<i>Information understandability</i> must be checked	Produce a Warning or Notification

575 these requirements. However, this evaluation by itself does not mean that the TranspLan language has quality, only models created by it. The quality of TranspLan as a modelling language will need further studies.

A generic framework has been developed for discussing the quality of models in general [40, 41]. This framework, called SEQUAL, focuses on a modelling quality as a means to achieve models with high quality [42]. The main concepts used in this framework are as follows:

- 580 • G is a set of organisational goals of the modelling task. In TranspLan, G denotes the

organisational goals related to providing meaningful and useful transparency to its relevant stakeholders through quality information. In other words, G denotes why this modelling of transparency requirements is being conducted in the information system.

- 585 • L is the set of all possible statements in a language according to graphemes, vocabulary, and syntax used in the modelling language. In TranspLan, L is defined by five sets of information elements, stakeholders, stakeholder-information relationships, decomposition relations, and information exchanges. These five sets and their corresponding symbols or graphemes in Shield diagram are defined and explained in TranspLan mathematically.
- 590 • D is the domain, or the set of all statements that can be stated about the situation at hand. In TranspLan, D denotes all possible information exchanges amongst various stakeholders which are involved in transparency provision and request.
- M is the externalised model, or the set of all statements in someone's model of part of the perceived reality written in a language. In TranspLan, M is defined based on a stakeholder's model of transparency expressed in TranspLan.
- 595 • K is the relevant explicit knowledge of the set of stakeholders involved in modelling. In TranspLan, K denotes the related knowledge of a stakeholder on TranspLan modelling language.
- I is the social actor interpretation, or the set of all statements that the audience thinks an externalised knowledge consists of. In TranspLan, I illustrates a stakeholder's interpretation of transparency requirements modelling in TranspLan.
- 600 • T is the technical actor interpretation, or the statements in the model as interpreted by different model activators or modelling tools. In TranspLan, T is the interpretations made by analytical reasoning on TranspLan.

Based on these definitions, quality types are defined as follows:

- 605 • **Physical quality** relates to the basic quality goals on the physical level that are externalised, that the knowledge K of the domain D of some social actor has been externalised by the use of a modelling language and internalised, and that the externalised model M is persistent and available, and so the audience can make sense of it.
- 610 • **Empirical quality** deals with predictable error frequencies identified when a model is read or written by different users through coding and visualisation, and by HCI-ergonomics for documentation and modelling tools.
- **Syntactic quality** is the correspondence between the model M and the language extension L of the language in which the model is written.
- 615 • **Semantic quality** is the correspondence between the model M and the domain D . The framework contains two semantic goals: validity, meaning that all statements made in the model are correct relative to the domain, and completeness, meaning that the model contains all statements which are found in the domain.
- 620 • **Perceived semantic quality** is the similar correspondence between the audience interpretation I of a model M , their current knowledge K of the domain D , and what can actually be checked during quality control.

- **Social pragmatic quality** is the correspondence between the model M and the audience's interpretation of that correspondence I . Social pragmatic quality refers to the extent to which people understand the model.
- 625 • **Technical pragmatic quality** refers to the extent to which tools can be constructed to understand the models.
- **Social quality** aims to find agreement amongst audience members' interpretations (I).
- **Organisational quality** of the model corresponds to the premise that all statements in the model either directly or indirectly contribute to fulfilling the goals of modelling (i.e., organisational goal validity) and that all goals of modelling are addressed through the model (i.e., organisational goal completeness).
- 630

Based on the quality criteria discussed above, the quality of models created by TranspLan can be divided into observational quality types and non-observational quality types. The observational quality types must be evaluated by observing the usage of the language, while the non-observational quality types can be evaluated independent from its usage in a real-world scenario. The observational quality types are:

- 635

1. Empirical quality,
2. Social pragmatic quality,
3. Social quality, and
4. Perceived semantic quality.

640 The non-observational quality types are:

1. Physical quality,
2. Syntactic quality,
3. Semantic quality,
4. Organisational quality, and
- 645 5. Technical pragmatic quality.

In order to find out the observational quality types of models created by TranspLan, an empirical study was conducted, which will be discussed in the next subsections. After that, the non-observational quality types of models created TranspLan will be discussed following the method used for evaluating the quality of UML modelling language [41]. However, we acknowledge that a complete study of the quality of the TranspLan language would require previous experience with the language so that the practitioners, participants, analysts, and researchers can reflect on their previous experience and practically evaluate the language at the meta-modelling level, similar to the applications of SEQUAL to evaluate UML [41]. In our evaluation, and given the constraints and the newness of the language, we took a bottom-up approach rather than the top-down one and tried to demonstrate, as a proof of concept, the quality of models created TranspLan through the quality of models and scenarios produced for it in a case study. Our evaluation aims to identify the participants acceptance and perception of TranspLan's ease of use and areas which are error-prone, in order to refine the language accordingly.

- 650
- 655

4.1. Case Study Aim and Questions

660 The aim of conducting this evaluation case study was to assess the extent of quality of models created by TranspLan in providing a systematic and effective engineering approach for the modelling and analysis of transparency requirements in an information system. In particular, the case study aims at finding how TranspLan is systematic and effective in the way it helps software engineers in the modelling and analysis of stakeholders' transparency requirements and in
665 the way it helps software engineers to be able to spot and elicit transparency requirements of stakeholders.

Following the Yin's approach [43] on the design of case studies, and based on the case study aim, the case study question can be formulated as follows:

670 **To what extent can models created by the TranspLan modelling language be comprehended by the participants to have empirical quality, social pragmatic quality, social quality, and perceived semantic quality?**

This evaluation case study has the following propositions:

- Models created by the TranspLan modelling language have certain qualities that can be systematically evaluated.
- 675 • These qualities make it suitable for TranspLan to be utilised by software engineers in the engineering of transparency requirements.

4.2. Study Design and Conduct

In order to conduct the case study in the second part of the evaluation of TranspLan, it was divided into an observational and a non-observational evaluation. For the observational part,
680 which is the empirical part of the evaluation, 12 software engineers were recruited, with a minimum of three, a maximum of 14, and an average of 6.08 years of experience in system analysis and design (See Figure 2). They also defined their modelling skills as fair (4 people), good (6 people), or very good (2 people). The unit of analysis in this case study was therefore the collection of software engineers in the empirical evaluation of TranspLan modelling language
685 quality. Participants were initially chosen based on their software system analysis and design (SSAD) experience. The average of more than six years of experience in SSAD meant that participants were familiar with SSAD concepts, including modelling. The participants mostly classified themselves as having good modelling skills (e.g., having worked with UML, BPMN, and Goal Modelling).

690 The details of a plan were laid early in the study in order to estimate the time, prepare the props and cater for the modelling session needs. The participants were given the modelling language and its theoretical foundations *one week* before the start of the study, and they were also briefed in a *half-hour session* about the modelling language. Furthermore, and to ensure everyone is familiar with the modelling language in practice, the study session also started with an
695 *introduction to the modelling language*, where the modelling was discussed and a small modelling task was given for them to complete. Then one possible solution was shown and discussed in the group.

The case study contained four parts. In the first part of the study, 6 participants were asked to draw a model based on the given scenario, while 6 people were asked to detail a scenario
700 based on a given model. Then, the two groups shifted their tasks, i.e., the first group started to

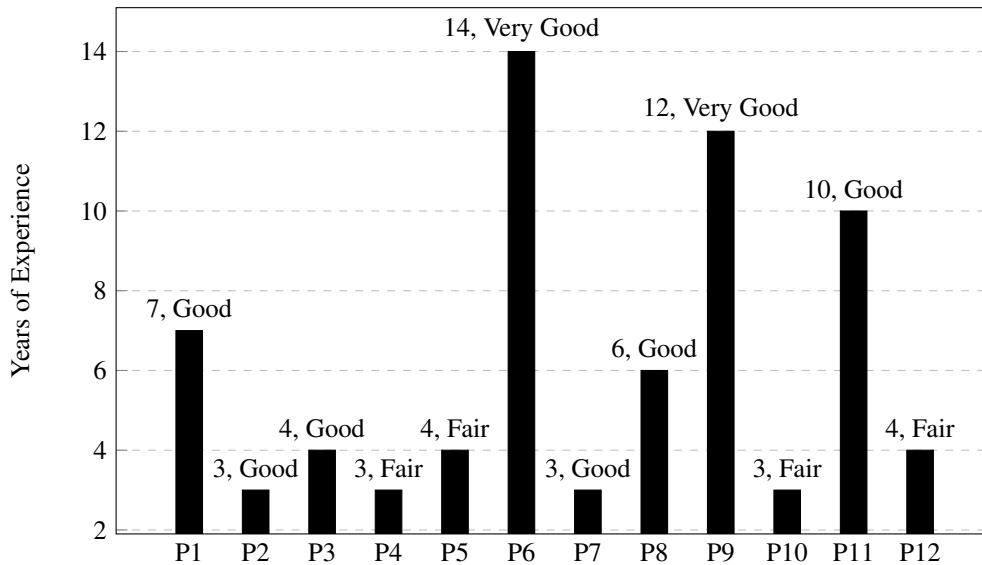


Figure 2: Experience years and modelling skills of the participants

build a scenario while the second group started to draw a model. This method was selected in order to reduce the learning effect [44]. The learning effect does not happen if the participant is only exposed to one condition and does not learn from a previous task. In the second part, every participant was given another participant's model and scenario to evaluate. This method was selected in order to remove the bias of the investigator from the evaluation part and also to measure the social quality and social pragmatic quality criteria. In the third part, a questionnaire was given to the participants to answer. The questions covered questions on their evaluation as well as general questions on TranspLan. In the fourth and last part, a discussion was held in order to hear participants' opinions and suggestions, as well as discuss their modelling experience. The whole session took 2 hours and 30 minutes to complete, and it was audio recorded.

In order to avoid complications during the evaluation, such as time pressure on participants and possible loss of interest during the evaluation which could lead to hasty answers, and to minimise the overhead on the participants given the various modelling tasks, we chose moderately complex scenarios. The scenarios were chosen from relatively common and well-understood domain such as banking and email service provision. The rationale was to be focused on the particular quality aspects of the models created by the language and isolate the side-effects which may originate from the scale of the scenario and its domain peculiarities. Participants did not use modelling tools because one of the tests we wanted to conduct was about the clarity of the syntax and whether they would make mistakes in this aspect. Using a modelling tool would not allow us to test that.

Participants did not know whose model they were evaluating as our instructions did not ask participants to put their names on the models. Furthermore, the evaluators were neither interested nor given those names. We acknowledge that during the discussion part, the originator of a model sometimes became clear, but this was after they had given their quantitative answers on the survey.

In order to avoid unnecessary complications during the evaluation, such as time pressure on

participants and possible loss of interest during the evaluation which could lead to hasty answers, and to minimise the overhead on the participants given the various modelling tasks, we chose relatively moderate scenarios. Participants did not use modelling tools because one of the tests we wanted to conduct was the clarity of the syntax and whether they would make mistakes in this aspect. Using a modelling tool would not allow us to test that.

4.3. The Scenario and the Task of Model Building

Participants were given the following scenario and were asked to model transparency provision and requests from the customer relationship management (CRM) viewpoint (i.e., with the CRM as the information provider). They were given a TranspLan Quick Reference Card (similar to Figure 1) for the ease and speed of modelling:

Scenario: A financial institution plans to introduce several improvements to their current Customer Relationship Management (CRM) plans. As part of the improvement, they are introducing customer categories based on existing customer information on their databases. The categorisation is meant to amend their existing transparency policies of the institution, providing information to customers in a way that minimises information overload and maximises their decision making power. The CRM management has identified three types of customers:

- Class A customers: they have at least four financial products (e.g., current account, saving account, ISA account, mortgage, Loans, credit card) with the institution. They are usually the most loyal customers, are very important and beneficial to the institution, and are very important for the financial institution to keep. Their informational needs must be always met, and they must be updated instantly with new products and services.
- Class B customers: they are customers with a current account and at most one more product. The institution policy is to encourage them to take up more new products with them, and increase their interaction and therefore increase their loyalty to the institution.
- Class C customers: they are ex-customers who have no financial services and products with the financial institution at the moment.

There are three information types usually communicated to these stakeholders:

- Information on new products and services: This information is mainly communicated to class B customers to encourage them to engage more with the institution and to increase their loyalty. This information is available to class A customers only on demand. The provision of this information is legal or coercive.
- Information on updates to existing products and services: This information is mainly communicated to Class A customers, but also to class B customers. The provision of this information is optional.
- Information on Xclusive Club benefits: The new scheme of the institution (Called Xclusive Club) is a legal necessity to provide an exclusive club with exclusive benefits (such as high cash back rates and holiday flights) only to Class A customers, and this must not be communicated to class B or class C customers.

Furthermore, the CRM management has decided to provide the following information to customers, based on legal demands:

- Savings and investment newsletter: including process and policies, and available on demand (i.e., subscription via institution website) to all customers except Class C.
- Institution annual financial report: To everyone (customer or otherwise) with an access to the website.

4.4. The Model and the Task of Scenario Building

Participants were also given the following model (Figure 3) and were asked to write a scenario based on it. They were asked to write their scenarios for each information exchange between two stakeholders and to include as much information as they could observe in the model.

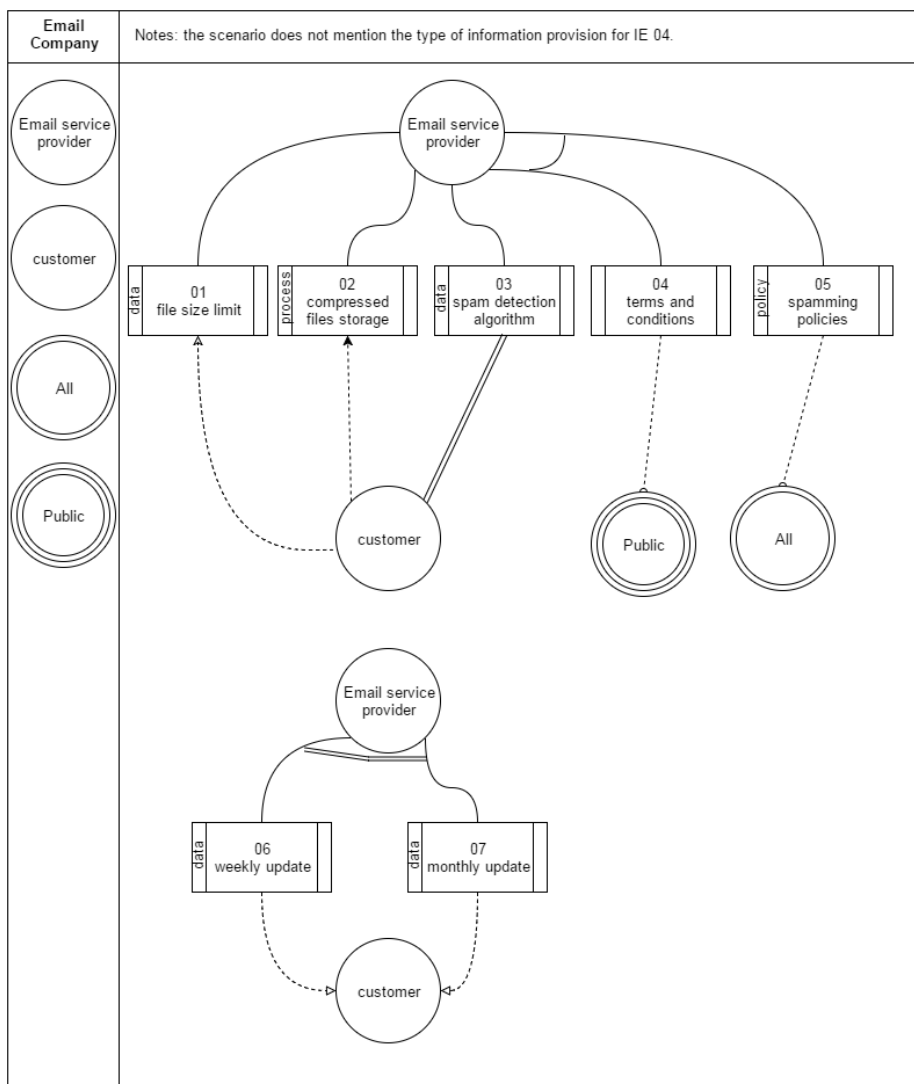


Figure 3: Case study model for participants to write a scenario from

775 4.5. *Questionnaire: Evaluation of TranspLan*

The questionnaire consisted of four parts. The first two parts assessed the participants' evaluation of another participant's drawn transparency model (hereby called the given model) and written scenario (hereby called the given scenario), while the last two parts assessed the participants' evaluation of the transparency model and scenario building in general. In this section, 780 the results obtained from these four parts are discussed, and conclusions are drawn upon these results at the end of each subsection.

4.5.1. *Evaluating the Given Model*

Procedure: The evaluation of another participant's model was conducted in order to find similarity amongst participants in their modelling practice, which can be translated into social 785 quality. It was also conducted to investigate whether the participants could understand different models, which can be translated into social pragmatic quality, and also to find out whether participants could highlight issues and problems with models drawn by other participants, which can be translated into empirical quality.

In order to find similarities amongst models, three measurement methods are proposed [45]:

- 790 1. Syntactic similarity, where only the syntax of the models are considered,
2. Semantic similarity, where the syntax is abstracted from the models, and the semantics of the words within the models are investigated, and
3. Contextual similarity, where the context in which the models occur is explored.

The questions in this part targeted the correctness of the model under evaluation in addition 795 to the similarity between the models. Since the twelve participants were well-trained to use the modelling language and depict models for a given scenario, it would be realistic to assume that they would see their modelling as generally correct or containing minor modelling flaws. This would be particularly true given the relatively clear scenario given to them and its common nature which does not require specific domain knowledge. In this case, if they considered the 800 model they were evaluating to be similar to their own model, it would be highly unlikely for them to think of that as an incorrect one. Such a consideration would imply that they also believed that their model was at least partially incorrect. This was not only an assumption we made but indeed the type of behaviour that we actually observed during the evaluation session. A report of incorrectness was accompanied by a report of dissimilarity between the model drawn 805 by the evaluator and the model drawn by the evaluatee. We here acknowledge the potential of having some slight differences, e.g., in linguistics and terms used or in the refactored shape of the model, which can enlighten the evaluator to detect errors in their original models despite the general similarities. However, we have not encountered that in the study, perhaps because of the relatively small-scale nature of the scenario and the sample.

Results: Regarding the *syntax* of the models, participants thought everybody else followed 810 the correct syntax of the modelling language, with all 12 participants agreeing with the syntactic correctness of the model under evaluation. They also agreed (11 participants agreeing and one remaining neutral) that there was a syntactic similarity between their model and the one they were evaluating. These results indicate social quality in terms of modelling syntax.

815 Regarding the *semantics* of the models, participants thought everybody else was following the correct semantics during their modelling, with 11 participants agreeing with the semantics correctness and one remaining neutral. Out of these 11, ten participants agreed with the semantics

similarity between their model and the one they were evaluating. These results indicate social quality in terms of the modelling semantics.

820 Finally, regarding the *contextual* similarity, most participants agreed that the model they were evaluating represented the scenario they were given, with seven participants agreeing with the contextual correctness and two participants remaining neutral. The same nine mentioned participants also agreed that there was a contextual similarity between their model and the one they were evaluating. These results indicate social quality in terms of modelling context.

825 The results of the evaluation in this part of the study are summarised in Table 4.

Table 4: Summary of the results on model correctness and similarity

	Syntactic Correctness	Syntactic Similarity	Semantic Correctness	Semantic Similarity	Contextual Correctness	Contextual Similarity
(Strongly) Agree	12	11	11	10	7	8
Neutral	0	1	1	1	2	1
(Strongly) Disagree	0	0	0	1	3	3

These results highlighted three types of quality found in the models created by TranspLan modelling language:

1. Models created by TranspLan have social pragmatic quality, i.e., the extent to which people understand the model, since they can judge whether a model is correct or not.
- 830 2. Models created by TranspLan have social quality, i.e., the amount of agreement amongst audience members' interpretations, since they judged an overall similarity between their model and another participant's model. The disagreements are in line with correctness of the evaluated model.
- 835 3. Models created by TranspLan have empirical quality, i.e., predictable error frequencies identified when a model is read or written by different users through coding and visualisation, since the participants could point out problems with the models under evaluation.

840 It is worth mentioning that our stated qualities for models created by TranspLan in this subsection are an objective reporting of the answers of the participants of this particular study. In other words, we report the perceptions of the participants regarding the models created by this modelling language, which is a commonly accepted analysis and reporting approach in empirical software engineering.

4.5.2. Evaluating the Given Scenario

845 **Procedure:** The evaluation of another participant's scenario was conducted in order to find out whether participants could highlight issues and problems with scenarios written by other participants, which can be translated into empirical quality. It was also conducted to investigate whether back translation or round-trip translation (i.e., drawing the same or similar model based on the written scenario) is possible [46] and whether the scenario under evaluation is useful. By usefulness of the scenario, here it is meant that the scenario under evaluation reveals transparency actors (i.e., the information provider and the information receiver), the disclosed

850 information and its type (i.e., data, process and policy), and the type of information disclosure
(e.g., optional/personal and coercive/legal disclosure).

Results: Regarding the *correctness* and *similarity* of the scenarios, most participants agreed that the scenario they were evaluating represented the model they were given, with eight participants agreeing with the scenario correctness and two participants remaining neutral. The same
855 ten mentioned participants also agreed that there was a similarity between their scenario and the one they were evaluating.

On the topic of *back translation*, participants showed divided opinions, with half of them agreeing that back translation is possible, but four remaining neutral and two disagreeing. The reasons could be linked to studies which suggest any translation from one language (here, the
860 transparency model) into another (here, the written language) loses certain characteristics and meanings [47], which then makes it difficult to translate back with the same amount of meaningfulness into the original one.

Regarding the *usefulness* of the scenario, participants generally found the scenario they were evaluating useful in terms of the identification of the constituents of transparency, with eight
865 participants agreeing and four participants remaining neutral. This illustrates that the scenarios under evaluation could be used by software engineers for transparency elicitation reasons.

The results of the evaluation in this part of the study are summarised in Table 5.

Table 5: Summary of the results on scenario evaluation

	Scenario Correctness	Scenario Similarity	Back Translation	Scenario Usefulness
(Strongly) Agree	8	7	6	8
Neutral	2	3	4	4
(Strongly) Disagree	2	2	2	0

The results obtained from this part of the evaluation highlighted one type of quality found in the models created by the TranspLan modelling language, plus the level of usefulness of the scenarios and their back-translation capabilities:
870

1. Models created by TranspLan have empirical quality, i.e., predictable error frequencies identified when a model is read or written by different users through coding and visualisation, since the participants could point out problems with the scenarios under evaluation.
2. Scenarios which were evaluated were useful, helping the participants identify transparency constituents.
875
3. There was uncertainty amongst some participants whether back translation would be accurately possible with the given scenario.

It is worth mentioning again that our stated qualities for models created by TranspLan in this subsection are an objective reporting of the answers of the participants of this particular
880 study. In other words, we report the perceptions of the participants regarding this modelling language, which is a commonly accepted analysis and reporting approach in empirical software engineering.

4.5.3. General Evaluation of TranspLan Modelling Language

Procedure: In this part of the evaluation, the TranspLan modelling language was evaluated independently from the drawn models or written scenarios and on its own merit. For this evaluation part, two sets of questions were asked. The first set of questions, including four questions, investigated ontological completeness and ontological clarity [48]. A modelling language is ontologically complete if all real-world concepts that should be captured by that modelling language can be represented by it. Otherwise, the modelling language is ontologically incomplete, or said to have construct deficit. A modelling language is ontologically clear if it has no construct overload, no construct redundancy, and no construct excess.

Construct overload happens in a modelling language when one design construct maps into two or more real-world concepts. For example, if in a modelling language a rectangle is used to represent a process and also to represent an actor, then the modelling language has construct overload. Construct redundancy occurs in a modelling language when two or more design constructs can be used to represent a single real-world concept. For example, if in a modelling language a rectangle and a circle are used to represent a process, then the modelling language has construct redundancy. Construct excess arises in a modelling language when there is a design construct that does not map into any real-world concept. For example, if in a modelling language an arrow type is never used to represent a concept, then the modelling language has construct excess. The existence of any of these three issues (i.e., construct overload, construct redundancy, and construct excess) leads to ontological ambiguity in a modelling language [48].

The participants in the study were software engineers and were familiar with the evaluation terminology used in this study. To assert that, however, we explained the evaluation terminology at the beginning of the evaluation session, gave examples about SEQUAL dimensions, and also elaborated on ontological aspects. The participants were also given the chance to clarify that with the facilitators before and during the study.

The second set of questions in this part of the evaluation investigated six evaluation criteria. These six criteria are *expressiveness*, *frequency of errors*, *redundancy*, *locality of change*, *reusability*, and *guidelines* [49]. In a modelling language, expressiveness refers to both the possibility and the ease of expressing real-world concepts and to effectively conveying the meaning of that concept. The existence of frequency of error means that the constructions in a modelling language are error prone and these errors can happen often during the modelling activity. Redundancy refers to construct redundancy which was explained above. Locality of change means that changes in one part of the model do not propagate to the other parts of the same (or another) model. Reusability denotes that fragments or the entire model can be used with no or little modifications in another model. Finally, the existence of useful and comprehensive guidelines helps the modellers to draw models with comfort and confidence, and get help from these guidelines when needed.

Results: In this evaluation parts, all participants agreed that TranspLan has no construct overload, which means that every modelling construct in TranspLan can be used for only one concept. They mostly agreed that TranspLan has no construct redundancy, meaning that one concept can be modelled with only one modelling construct. The only person who pointed to construct redundancy in TranspLan pointed out that “All Actors Except” construct can be equal to a set of actors. While this is the case, the reason for devising this construct is to make the model less cluttered during the design process. Furthermore, participants found no construct excess in TranspLan, meaning that all modelling constructs in TranspLan have a real-world corresponding concept. They mostly agreed that TranspLan, within its remit and goals, did not have construct

930 deficit while they tried it on the scenarios. This implies that participants perceive TranspLan to have constructs that can model the transparency aspect of a business information system. The only person who pointed to construct deficit stated that the language does not capture whether the requested information is actually provided to the information receiver or not. While this is not the initial concern in TranspLan, one way to address this issue is to use solid lines for ‘required information provision’ and dotted lines for ‘unrequired information provision’. The answers are summarised in Table 6 and Figure 4.

Table 6: A summary of answers given to the absence of ontological clarity and completeness in TranspLan

	Absence of Ontological Clarity			Absence of Ontological Completeness
	Construct Overload	Construct Redundancy	Construct Excess	Construct Deficit
Yes	0	1	0	1
No	12	8	11	7
IDK	0	3	1	4

940 While participants agreed on several ontological aspects of the TranspLan modelling language, we would like to point out that these conclusions are solely based on the perceptions of the participants and before any firm conclusions are to be made, other in-depth studies should be carried out. As such, this evaluation serves only as a pilot study towards a more concrete ontological evaluation of TranspLan.

945 With regards to *expressiveness*, participants generally agreed that TranspLan is expressive, meaning that the modelled concepts effectively convey the meaning of that concept. The only person who found an example of lack of expressiveness in TranspLan pointed to the “undecided” relationship type, which they considered to be ambiguous in conveying the meaning of the provision type. However, this type of relationship was intentionally put in TranspLan to capture instances of transparency where the information provision type is unknown at the time of design. Therefore, it is another example of expressiveness and ontological completeness of TranspLan.

950 With regards to the *frequency of error*, half of the participants agreed that certain modelling mistakes do not happen several times, while three participants disagreed and mentioned that certain modelling mistakes might happen several times during modelling with TranspLan. These three people all pointed out that the arrow direction and head type can become confusing, while one also mentioned that “All Actors Except” construct could be equal to “restricted” construct for those stakeholders who are exempt from information provision, and therefore could be confusing. In response to the first concern, the arrowheads type and direction can be mastered by practice, as is the case in other modelling languages with several similar constructs, such as the gates types in BPMN. Furthermore, “All Actors Except” construct does not mean that stakeholders who are exempt from information provision are banned from getting such information, only that they are not simply the relevant stakeholders for that information, which is different in meaning from “restricted” stakeholders, who are actually banned from getting such information.

955 With regards to *locality of change*, all participants agreed that changes in TranspLan are local, i.e., changing part of the model does not propagate and does not require changes in other parts of the model to maintain its consistency and correctness.

With regards to *reusability*, most participants agreed that TranspLan has the reusability feature, meaning that parts or the entire model in one scenario may be used with little customisation in another scenario. For example, one participant pointed out similar scenarios related to customer support in ticketing systems in buses, trains and flights where TranspLan could be reused, while another participant pointed to similarities of information exchange in many environments regarding Terms and Conditions documents, where TranspLan could benefit from reusability.

With regards to *guidelines* provided for TranspLan, half of the participants agreed that the information that was provided in the conference paper on the TranspLan modelling language [8] and the guidelines and reference guides given to them during the study session were complete enough for drawing and understanding the model drawn in TranspLan. The two people who answered otherwise referred again to another issue not necessarily related to guidelines. For example, one participant asked whether information becomes available when it is requested by a stakeholder, which they could not find in the guidelines, and the answer to which, as explained earlier, is implied to be positive in the modelling language at the moment. However, this could be further clarified in the next version of the language as well. The other participant also mentioned that when information is surely received by the information receiver, there is no way to model it using the modelling language. The same response in the previous example also applies to this comment. The results of the evaluation in this part of the study are summarised in Figure 4.

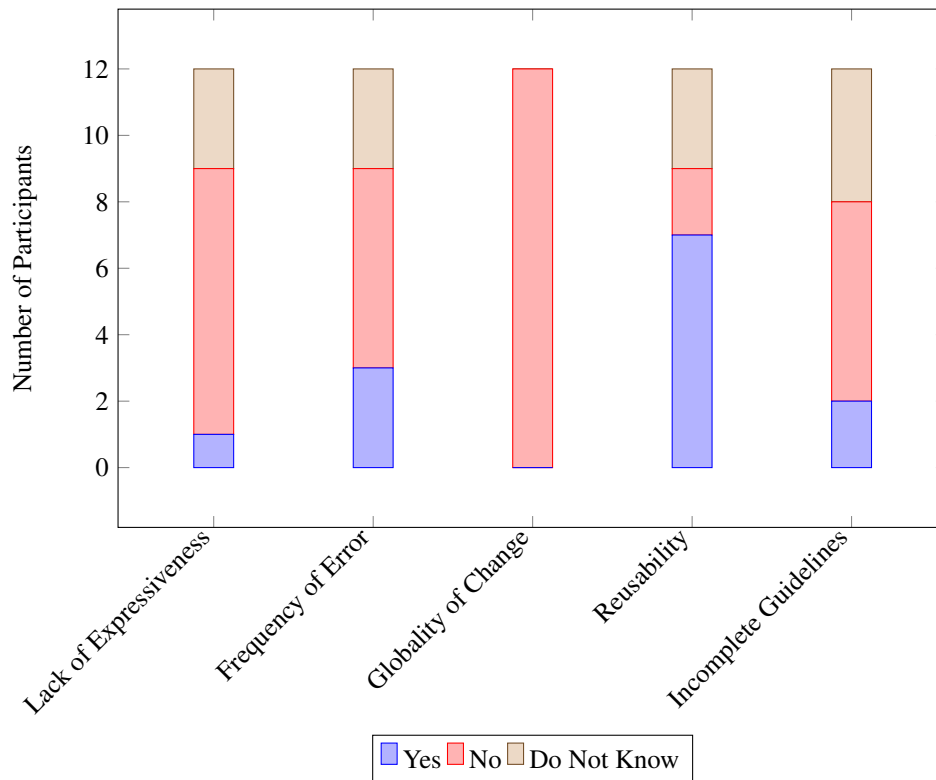


Figure 4: Summary of answers to questions in part three

The results obtained from this part of the evaluation highlighted the following outcomes:

1. Models created by TranspLan have perceived semantic quality, because it is perceived to be ontologically clear and complete, with no issues found during the empirical evaluation of the TranspLan modelling language. In part, this could reflect the syntactic and semantic quality of TranspLan.
2. TranspLan is expressive, with little or no frequency in making errors, and has the reusability characteristics. Furthermore, changes in TranspLan are local and the guidelines provided for it are complete.

It is worth mentioning again that our stated qualities for models created by TranspLan in this subsection are an objective reporting of the answers of the participants of this particular study. In other words, we report the perceptions of the participants regarding this modelling language, which is a commonly accepted analysis and reporting approach in empirical software engineering.

4.5.4. General Evaluation of Scenario Building in TranspLan

Procedure: In this last part of the evaluation, the TranspLan modelling language was evaluated in terms of scenario building based on drawn models. The logic in this part was that a model should be easy enough to extract from a drawn model of transparency, the extracted scenario should be expressive in identifying the constituents of transparency, it should not be prone to frequent errors, and the guidelines should be complete enough in the act of scenario building. These were the four questions which participants answered in their evaluation of scenario building in TranspLan.

Results: With regards to *scenario extraction*, participants generally agreed that building a scenario is easy based on a given model, with ten participants agreeing and one participant remaining neutral. This means that the model is easy to read and is therefore another sign of social pragmatic quality of the TranspLan modelling language.

With regards to *scenario expressiveness*, all participants agreed that the scenario built based on a model is expressive and conveys the meaning of concepts in that model, e.g., effectively reveals transparency actors. This is yet another sign of social pragmatic quality of TranspLan.

With regards to the *frequency of errors*, half of the participants pointed out that making a mistake several times does not usually occur while building a scenario based on a model drawn using TranspLan. On the other hand, four participants thought it is possible to make a certain mistake several times during scenario building. This again was mostly the result of interpreting arrowheads type and direction, which can be mastered over time.

With regard to the provided *guidelines* for scenario extraction, all participants agreed that the guidelines provided in the conference paper on TranspLan modelling language [8] and the guidelines and reference guides given to them during the study session were complete enough for building a scenario based on a given model drawn using TranspLan.

The results of the evaluation in this part of the study are summarised in Figure 5.

The results obtained from this part of the evaluation highlighted the following outcomes:

1. Models created by TranspLan have social pragmatic quality, since participants could understand the model and could build and extract expressive scenarios based on it.
2. TranspLan is expressive in scenario building, with little or no frequency in making errors. Furthermore, the guidelines provided for TranspLan are complete and help in scenario extraction.

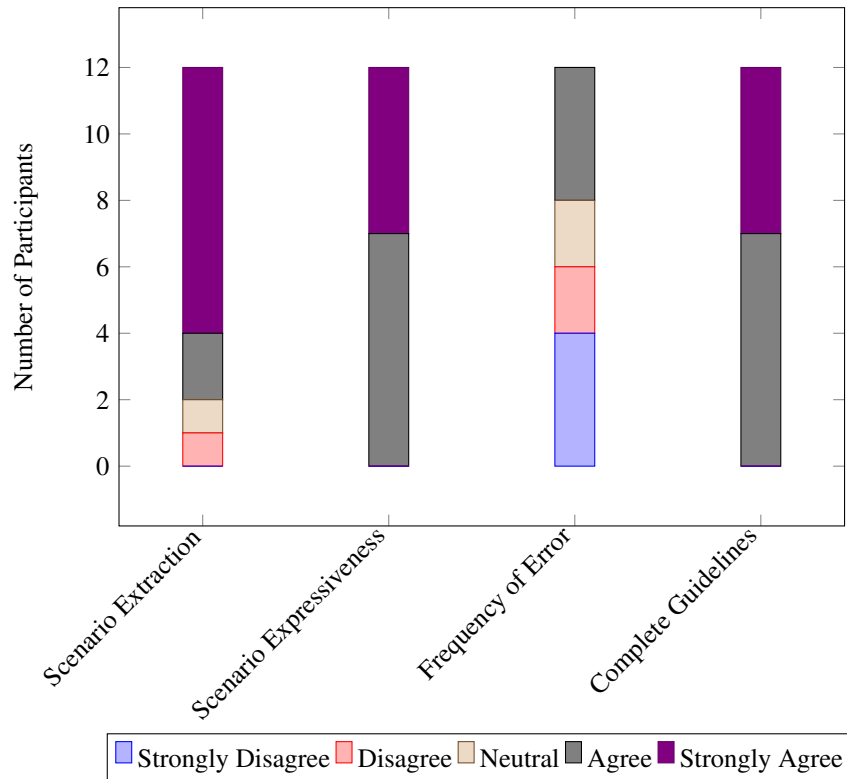


Figure 5: Summary of answers to questions in part four

1025 It is worth mentioning one last time that our stated qualities for models created by TranspLan
 in this subsection are an objective reporting of the answers of the participants of this particular
 study. In other words, we report the perceptions of the participants regarding this modelling
 language, which is a commonly accepted analysis and reporting approach in empirical software
 engineering.

1030 4.6. TranspLan and Quality Criteria

The empirical study which was conducted helped identify four types of quality criteria which
 were categorised as observational quality types. In other words, the empirical study suggested
 that models created by TranspLan can have empirical quality, social pragmatic quality, social
 quality, and perceived semantic quality.

1035 In this subsection, the non-observational quality types will be investigated. The aim is to
 identify whether TranspLan supports the assessment of these non-observational quality types for
 individual models.

4.6.1. Physical Quality of Models Created by TranspLan

1040 Physical quality relates to the basic quality goals on the physical level that are externalised,
 that the knowledge K of the domain D of some social actor has been externalised by the use

of a modelling language and internalised, and that the externalised model M is persistent and available, and so the audience can make sense of it.

Physical quality has two main aspects. The first one is the externalisation of the knowledge using the modelling language. It can be argued that TranspLan externalises the knowledge on transparency because it is based on the reference models which capture several (if not all) aspects of transparency requirements. That is to say, TranspLan has the capability to externalise the knowledge on transparency so that people can make sense of it and discuss it. In terms of documentation, the specifications of TranspLan is primarily externalised as a documentation which includes examples.

The second aspect is the internalisation of the knowledge. How people perceive the model is a matter of internalisation. A persistent and available modelling language helps the internalisation process. TranspLan is available for everyone and has only one version and one interpretation at the moment; therefore it is persistent as well. This helps the internalisation process of TranspLan. Consequently, it can be concluded that models created by TranspLan can have physical quality.

4.6.2. *Syntactic Quality of Models Created by TranspLan*

Syntactic quality is the correspondence between the model M and the language extension L of the language in which the model is written.

TranspLan uses mathematical definitions for its constituents which gives it a vigorous structure. The syntax used in TranspLan is detailed to the understanding level of its users and the examples given are according to the syntax of TranspLan. The three-layered structure of each information exchange facilitates easier error detection and error correction. Consequently, it can be concluded that models created by TranspLan can have syntactic quality.

4.6.3. *Semantic Quality of Models Created by TranspLan*

Semantic quality is the correspondence between the model M and the domain D . The framework contains two semantic goals: validity, meaning that all statements made in the model are correct relative to the domain, and completeness, meaning that the model contains all statements which are found in the domain.

In TranspLan, it was already explored empirically that the language benefits from ontological clarity and completeness. Furthermore, it can be argued that TranspLan makes all statements made in the model correct and related to transparency and also all statements found in the domain are contained in the model as it is founded on the reference models. This means that the two semantics goals of validity and completeness are achieved. Finally, the descriptions of the notation and semantics are fairly complete and there are no inconsistencies in the language. Consequently, it can be concluded that models created by TranspLan can have semantic quality.

4.6.4. *Organisational Quality of Models Created by TranspLan*

Organisational quality of the model corresponds to the premise that all statements in the model either directly or indirectly contribute to fulfilling the goals of modelling (i.e., organisational goal validity) and that all goals of modelling are addressed through the model (i.e., organisational goal completeness).

In TranspLan, all the statements in the model aid the goals of the modelling, meaning that every statement serves its purpose in identifying one aspect of transparency. This fulfils the organisational goal validity. Furthermore, all the goals of modelling (i.e., finding transparency meaningfulness, transparency usefulness, transparency stakeholders, and information quality in

transparency) are addressed through the model. This fulfils organisational goal completeness. Consequently, it can be concluded that models created by TranspLan can have organisational quality.

4.6.5. *Technical Quality of Models Created by TranspLan*

Technical pragmatic quality refers to the extent to which tools can be constructed to understand the models.

As it was shown in the previous chapters, TranspLan facilitates the development of analytical reasoning and the automated analysis of transparency requirements. The language also has the potential to be represented by Computer-Aided Software Engineering (CASE) tools for the modelling of transparency requirements. Consequently, it can be concluded that models created by TranspLan can have technical quality.

4.7. *Evaluation Conclusion*

Based on our evaluation of TranspLan, it can be concluded that this modelling language supports the creation of transparency models with the quality measures inspired by the SEQUAL framework. We also acknowledge that the quality of the language itself will need more studies and real-life applications before any definite conclusions can be made.

5. Discussion

In the light of the obtained results, the study question can now be answered as follows:

- **To what extent can models created by the TranspLan modelling language be comprehended by the participants to have empirical quality, social pragmatic quality, social quality, and perceived semantic quality?** The results of the study illustrate that models created by TranspLan have been perceived by the participants to have a good capability in modelling important aspects of transparency requirements in a business information system. The investigation into non-observational quality types (i.e., physical quality, syntactic quality, semantic quality, organisational quality, and technical quality) complements the evaluation case study and denotes that models created by the TranspLan modelling language may also benefit from quality types inspired by the SEQUAL framework. The obtained results imply that participants view TranspLan to be able to help software engineers in engineering transparency requirements with minimal difficulty when it comes to the quality of the models created by the language.

It should be noted that the results obtained from this case study are not definitive. Furthermore, in order to observe and evaluate the quality of TranspLan itself, it should be first used in a larger scale by the requirements engineering community so that it is verified across several real-world transparency modelling and analysis activities. This, of course, is not limited to TranspLan and modelling languages generally have to undergo this process. It is also understandable that TranspLan is a newly devised modelling language and consequently, the results of this evaluation case study remain valid until future usages.

5.1. Threats to Validity

In the empirical part of this study, twelve software engineers participated in the evaluation. The study session included four sections which together served to evaluate TranspLan in terms of expressiveness, locality of change, etc., while they also served to investigate four different types of quality criteria for the models created by TranspLan. In the empirical part of this study, some threats to validity were identified. These threats are listed below:

- It is well understood that the quality of a given modelling language can be assessed more realistically with real applications and implementations in which an organisation's transparency requirements are elicited, modelled, and analysed in a real-world setting. Only through the constant use of the modelling language in authentic requirements engineering scenarios can that modelling language be really tested and its qualities recognised. However, a case study approach would still be desirable for a newly-devised modelling language, and could provide a useful means to evaluate and assess its fundamental quality attributes.
- During the study, each participant could only evaluate one model and scenario of another participant. This could lead to conditions where bias against a participant could affect the evaluation of their models and scenarios. However, models and scenarios were randomly assigned to participants to minimise the effects of such bias.

5.2. Lessons Learnt

This study highlighted the quality criteria associated with models created by TranspLan and its potential for being used as a modelling language for the engineering of transparency requirements in information systems. However, during the discussion on the modelling and scenario building with TranspLan, some interesting comments were also observed.

Participants appreciated the three-layered structure of the information exchanges in TranspLan, as it made it easy for them to draw them once they learned the basics of the language. According to one participant, it also made it easier for them to avoid possible errors during modelling and detect and correct them in their (or other people's) models.

With regards to the graphical representation, one participant mentioned that the drawing of two parallel lines (which indicates limited access to an information element) could be a bit un-gainly and a possible source of confusion when drawn by hand. They suggested an alternative, like a line with a cross over it instead. On the other hand, participants appreciated the use of a few shapes which could increase the learning curve. For example, they appreciated the use of rectangles for all information types (i.e., data, process, and policy) instead of devising a shape for each one of them.

It was also observed during the evaluation session that drawing a model from a given scenario is more difficult than writing a scenario based on a given model. It can be argued that this is shared by several modelling languages, as writing in your own language takes less mental power than drawing in a modelling language, especially a new one, where you have to remember the meanings of shapes, lines, etc.

Finally, we acknowledge that it is important to consider the learning curve for the practitioners in using TranspLan as a visual modelling language. This aspect seems to be particularly relevant in the specific context of transparency since the concept of transparency for organisations and information systems is itself in transformation and clarification.

5.3. Enhancements to TranspLan

1165 Conducting this case study helped us develop a better understanding of the needs of software
 engineers from our modelling language. While the fundamentals of the language were deemed
 as vigorous (e.g., having ontological clarity and completeness, being expressive, and benefiting
 from locality of change), the participants pointed out a few enhancements to the TranspLan
 modelling language in terms of its graphical representation and visual construction. In particular,
 1170 the following three enhancements, as discussed earlier, were considered, based on which the
 graphical representation of TranspLan were modified.

Visual Syntax	Semantics	Visual Syntax	Semantics
<u>Stakeholders</u>		<u>Information Elements</u>	
	stakeholder <i>S</i>		information element <i>i</i> containing data
	all stakeholders in the diagram		information element <i>i</i> containing a process
	all stakeholders in the diagram (except <i>S</i> ₁ , <i>S</i> ₂)		information element <i>i</i> containing a goal
	public stakeholders		unspecified information element <i>i</i>
<u>Relationships</u>		<u>Stakeholder/Information Relations</u>	
	AND relationship		stakeholder <i>S</i> producing information <i>i</i>
	OR relationship		stakeholder <i>S</i> requesting information <i>i</i>
	exclusive OR relationship		stakeholder <i>S</i> receiving information <i>i</i>
			<u>information <i>i</i> not intended for stakeholder <i>S</i></u>
<u>Relation Types</u>		<u>Information Exchange System</u>	
	<u>Obligatory relationship: Coercive Provision / Legal Demand</u>		information exchange system boundary with the list of stakeholders
	<u>Optional relationship: Voluntary Provision / Personal Demand</u>		
	<u>Undecided relationship: Unknown Provision or Demand</u>		
<u>Medium Used</u>		<u>Information Requirement Types</u>	
	<u>Information received/requested without mentioning the medium</u>		<u>Unrequited information provision</u>
	<u>Information received/requested with mentioning the medium</u>		<u>Required information provision</u>

Figure 6: The modified graphical representation for the building blocks of *Shield* and their interpretations, with new modifications underlined in the Semantics sections

- 1175 • One concern expressed by participants were about whether a request for information is actually fulfilled or whether provided information is actually received by information receivers. We already stated that this was not an initial concern in TranspLan. However, to accommodate for this concern, we modified TranspLan by adding two types of lines: *solid lines* for ‘requited information provision’ and *dotted lines* for ‘unrequited information provision’, as can be viewed in Figure 6.
- 1180 • Another concern that participants mentioned was with regards to the drawing of two parallel lines for representing limited access to an information element. The concern was especially directed towards those model drawn by hands, where two parallel lines could be, at times, easily mistaken with two separate lines. To address this issue, we changed this representation to a simple line with a cross sign over it, as can be viewed in Figure 6.
- 1185 • The last concern stated by participants were the absence of information medium in acts of information exchange. In order to cater for this concern, a decision was made based on which every requested information or received information can be optionally augmented by the name of the information medium on the corresponding arrow, as can be viewed in Figure 6.

6. Conclusion and Future Work

1190 In this article, we presented a modelling language for transparency requirements, called TranspLan, and evaluated it. The first part of the evaluation was performed with software engineers and it incorporated two aspects of building a model based on a given scenario and extracting a scenario based on a given model, while also evaluating the TranspLan modelling language and its scenario building capabilities in general. The second part of the evaluation was based on the general characteristics of the TranspLan modelling language, its mathematical foundation, and
1195 the guidelines provided for it.

The results of the evaluation highlighted the existence of general modelling quality in several aspects of TranspLan. However, some observations were also made by the software engineers involved in the study about some graphical aspects of the TranspLan modelling language. These observations helped the creators of this modelling language to enhance the graphical representation of the TranspLan modelling language by introducing a few tweaks to it.
1200

As the TranspLan modelling language is a newly devised language for a relatively novel concept (i.e., the engineering of transparency requirements), several future works are possible. One example of such future work is to evaluate this modelling language in terms of its usability from a Human Computer Interaction (HCI) perspective. As a fundamental feature of a modelling language, usability of TranspLan can be the subject of a future work, which will give weight to its applicability and use for the modelling of transparency requirements. Furthermore, TranspLan can be used for assessing transparency requirements of stakeholders in other domains, such as intelligent environments [50].
1205

Another possible future work is to integrate TranspLan to mainstream modelling languages and methodologies or generate variations of it which can fit them. Besides integrating and empowering TranspLan with existing machinery, this is indeed conceptually needed as transparency is not a standalone concept but rather a complementary one to a hosting business information system where other business requirements and non-functional requirements, such as trust, security and privacy, strongly inter-relate with transparency. Examples of such languages include the
1210

1215 goal modelling and business process modelling languages as they emphasise the socio-technical
nature of the system where transparency resides in the first place.

Another future work can focus on building a meta-model for the TranspLan modelling lan-
guage. The meta-model can be used to investigate the quality of the modelling language itself,
while it can also be used to produce automated tools and support for software engineers work-
1220 ing on transparency requirements. The authors are currently working on this meta-model for
TranspLan.

Another future work is related to augmenting the TranspLan modelling language with adja-
cent concepts such as privacy and security. Even though the language is mainly concerned with
transparency requirements of its stakeholders, it is inevitable that privacy and security concerns
1225 should also be considered while providing transparency. Therefore, it remains a future work for
interested scholars to augment the modelling language in a way that takes security and privacy
requirements of these stakeholders in mind.

Acknowledgements.

The research is supported by an FP7 Marie Curie CIG grant (the SOCIAD project).

1230 **References**

- [1] D. Curtin, A. J. Meijer, Does transparency strengthen legitimacy?, *Information polity* 11 (2) (2006) 109–122.
- [2] S. M. Hosseini Moghaddam, *Engineering of transparency requirements in business information systems*, Ph.D.
thesis, Bournemouth University (2016).
- [3] M. Hosseini, A. Shahri, K. Phalp, R. Ali, Four reference models for transparency requirements in information
1235 systems, *Requirements Engineering* (2017) 1–25.
- [4] F. Bannister, R. Connolly, The trouble with transparency: A critical review of openness in e-government, *Policy &
Internet* 3 (1) (2011) 1–30.
- [5] F. Scauer, *Transparency in three dimensions*, *University of Illinois Law Review* (2011) 1339.
- [6] S. Ghanavati, D. Amyot, L. Peyton, Towards a framework for tracking legal compliance in healthcare, in: *Interna-
1240 tional Conference on Advanced Information Systems Engineering*, Springer, 2007, pp. 218–232.
- [7] D. C. Schmidt, *Model-driven engineering*, *Computer-IEEE Computer Society* 39 (2) (2006) 25.
- [8] M. Hosseini, A. Shahri, K. Phalp, R. Ali, A modelling language for transparency requirements in business infor-
mation systems, in: *International Conference on Advanced Information Systems Engineering*, Springer, 2016, pp.
239–254.
- [9] R. Araujo, Y. Taher, W. Heuvel, C. Cappelli, Evolving government-citizen ties in public service design and delivery,
1245 in: *IFIP EGOV Conference*, Citeseer, 2013, pp. 19–26.
- [10] M. Turilli, L. Floridi, The ethics of information transparency, *Ethics and Information Technology* 11 (2) (2009)
105–112.
- [11] S. L. Star, G. C. Bowker, L. J. Neumann, Transparency at different level of scale: convergence between information
1250 artefacts and social worlds, *Library and Information Science*, Urbana-Champaign.
- [12] P. Meunier, *Software transparency and purity*, *Communications of the ACM* 51 (2) (2008) 104–104.
- [13] J. C. S. do Prado Leite, C. Cappelli, Exploring i* characteristics that support software transparency., in: *iStar*, 2008,
pp. 51–54.
- [14] J. C. S. do Prado Leite, C. Cappelli, *Software transparency*, *Business & Information Systems Engineering* 2 (3)
1255 (2010) 127–139.
- [15] L. M. Cysneiros, Using i* to elicit and model transparency in the presence of other non-functional requirements:
A position paper., in: *iStar*, Citeseer, 2013, pp. 19–24.
- [16] L. Chung, B. A. Nixon, E. Yu, J. Mylopoulos, *Non-functional requirements in software engineering*, Vol. 5,
Springer Science & Business Media, 2012.
- [17] C. Cappelli, J. Sampaio do Prado Leite, A. d. P. A. Oliveira, Exploring business process transparency concepts, in:
1260 *Requirements Engineering Conference, 2007. RE'07. 15th IEEE International*, IEEE, 2007, pp. 389–390.
- [18] E. Yu, *Modelling strategic relationships for process reengineering*, *Social Modeling for Requirements Engineering*
11 (2011) 2011.

- 1265 [19] L. Dabbish, C. Stuart, J. Tsay, J. Herbsleb, Social coding in github: transparency and collaboration in an open software repository, in: Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, ACM, 2012, pp. 1277–1286.
- [20] L. Dabbish, C. Stuart, J. Tsay, J. Herbsleb, Leveraging transparency, *IEEE software* 30 (1) (2013) 37–43.
- [21] M. Serrano, J. Sampaio do Prado Leite, Capturing transparency-related requirements patterns through argumentation, in: Requirements Patterns (RePa), 2011 First International Workshop on, IEEE, 2011, pp. 32–41.
- 1270 [22] M. Serrano, J. do Prado Leite, J. Mylopoulos, Relative validation for transparency requirements, Tech. rep., Monograph, Department of Informatics, Pontifical Catholic University of Rio de Janeiro (February 2011).
- [23] H. Cunha, J. C. S. do Prado Leite, L. Duboc, V. Werneck, The challenges of representing transparency as patterns, in: Requirements Patterns (RePa), 2013 IEEE Third International Workshop on, IEEE, 2013, pp. 25–30.
- [24] S. T. de Oliveira, Ferramentas para o aprimoramento da qualidade, Pioneira, 1996.
- 1275 [25] M. Hosseini, A. Shahri, K. Phalp, R. Ali, Towards engineering transparency as a requirement in socio-technical systems, in: 2015 IEEE 23rd International Requirements Engineering Conference (RE), IEEE, 2015, pp. 268–273.
- [26] D. M. Strong, Y. W. Lee, R. Y. Wang, 10 potholes in the road to information quality, *IEEE Computer* 30 (8) (1997) 38–46.
- [27] M. K. Brunnermeier, Information leakage and market efficiency, *Review of Financial Studies* 18 (2) (2005) 417–457.
- 1280 [28] J. H. Morris, Tales of technology: Consider a cure for pernicious infobesity, *The Pittsburgh Post-Gazette*.
- [29] J. Ignacio Aguaded, From infoxication to the right to communicate (2014).
- [30] R. Savolainen, Filtering and withdrawing: strategies for coping with information overload in everyday contexts, *Journal of Information Science* 33 (5) (2007) 611–621.
- 1285 [31] D. Bawden, L. Robinson, The dark side of information: overload, anxiety and other paradoxes and pathologies, *Journal of information science* 35 (2) (2009) 180–191.
- [32] M. Lopp, Information starvation, *Managing Humans: Biting and Humorous Tales of a Software Engineering Manager* (2007) 33–36.
- [33] M. Vaughan, *The Thinking Effect: Rethinking Thinking to Create Great Leaders and the New Value Worker*, Nicholas Brealey Publishing, 2013.
- 1290 [34] D. Aboody, B. Lev, Information asymmetry, r&d, and insider gains, *The journal of Finance* 55 (6) (2000) 2747–2766.
- [35] Å. Blomqvist, The doctor as double agent: Information asymmetry, health insurance, and medical care, *Journal of Health Economics* 10 (4) (1991) 411–432.
- 1295 [36] J. Hultman, B. Axelsson, Towards a typology of transparency for marketing management research, *Industrial Marketing Management* 36 (5) (2007) 627–635.
- [37] A. Vaccaro, P. Madsen, ICT and an NGO: Difficulties in attempting to be extremely transparent, *Ethics and Information Technology* 11 (3) (2009) 221–231.
- [38] A. Meijer, Understanding modern transparency, *International Review of Administrative Sciences* 75 (2) (2009) 255–269.
- 1300 [39] J. Fox, The uncertain relationship between transparency and accountability, *Development in Practice* 17 (4-5) (2007) 663–671.
- [40] J. Krogstie, G. Sindre, H. Jørgensen, Process models representing knowledge for action: a revised quality framework, *European Journal of Information Systems* 15 (1) (2006) 91–102.
- 1305 [41] J. Krogstie, O. I. Lindland, G. Sindre, Defining quality aspects for conceptual models, in: *Information System Concepts*, Springer, 1995, pp. 216–231.
- [42] J. Krogstie, Evaluating uml using a generic quality framework, *UML and the Unified Process 1*.
- [43] R. Yin, *Case study research: Design and methods*. beverly hills (1994).
- [44] J. Lazar, J. H. Feng, H. Hochheiser, *Research methods in human-computer interaction*, John Wiley & Sons, 2010.
- 1310 [45] R. Dijkman, M. Dumas, B. Van Dongen, R. Käärrik, J. Mendling, Similarity of business process models: Metrics and evaluation, *Information Systems* 36 (2) (2011) 498–516.
- [46] H. Somers, Round-trip translation: What is it good for, in: *Proceedings of the Australasian Language Technology Workshop, 2005*, pp. 127–133.
- [47] F. Van Nes, T. Abma, H. Jonsson, D. Deeg, Language differences in qualitative research: is meaning lost in translation?, *European journal of ageing* 7 (4) (2010) 313–316.
- 1315 [48] Y. Wand, R. Weber, On the ontological expressiveness of information systems analysis and design grammars, *Information Systems Journal* 3 (4) (1993) 217–237.
- [49] F. Ruiz, F. van Harmelen, M. Aben, J. van de Plassche, Evaluating a formal modelling language, in: *International Conference on Knowledge Engineering and Knowledge Management*, Springer, 1994, pp. 26–45.
- 1320 [50] M. Amiribesheli, M. Hosseini, A. Bouchachia, A user-centred principle based transparency approach for intelligent environments, in: *Proceedings of Human Centred Design for Intelligent Environments Workshop at British HCI 2016*, 2016.