Modelling Metrics for Transparency in Medical Systems

Dayana Spagnuelo*, Cesare Bartolini, and Gabriele Lenzini

Interdisciplinary Centre for Security Reliability and Trust (SnT)
University of Luxembourg

Abstract. Transparency, a principle advocated by the General Data Protection Regulation, is usually defined in terms of properties such as availability, auditability and accountability and for this reason it is not straightforwardly measurable. In requirement engineering, measuring a quality is usually implemented by defining a set of metrics for its composing properties, but conventional approaches offer little help to achieve this task for transparency. We therefore review requirements for availability, auditability and accountability and, with the help of a meta-model used to describe non-functional properties, we discuss and advance a set of metrics for them. What emerges from this study is a better justified and comprehensive tool which we apply to measure the level of transparency in medical data-sharing systems.

Keywords: Transparency, Metrics, Availability, Auditability, Accountability

1 Introduction

Transparency is a principle that can be embraced to prove honesty and therefore trustworthiness. Applied in distributed data management systems, such as cloud computing, electronic banking, or medical data-sharing, transparency is also a strategic element in business. Personal data are an asset [16, 17] and the data providers which monetise on them may suffer the mistrust of data subjects (e.g., users, clients, patients) who, aware of the risks to expose personal information [8], can be reluctant to consent to the processing of their personal data. Transparency here can help build and preserve trust: providers that offer detailed information about their policies and practices, express them in a clear and readable manner, and have easily-accessible documents and histories of processing operations, will also have a better chance to gain their client's trust and stay in business.

There are also other reasons to implement transparency. The importance of being transparent has being spurred by the recent legal reform on data protection

* Supported by FNR/AFR project 7842804 TYPAMED

in the European Union[1]. In this context, transparency is intended to guarantee that the systems are processing personal data in a lawful and fair manner.

But how can transparency be modelled and implemented, and how to measure the amount of transparency that a system is able to guarantee?

The problem of *modelling and implementing transparency* in terms of a Requirements Engineering (RE) approach has been already explored [4, 20]. Transparency is described as a Non-Functional Requirement (NFR) that offers some degree of monitoring over the systems by providing the users with *information* and *mechanisms*. Both intended to impart knowledge on how the user's data has been or will be processed.

The problem of *measuring transparency* can also be tackled by following a RE approach. Assuming that transparency is translated into a set of specific requirements, measuring transparency means defining *metrics* to evaluate to what degree the requirements are met in a system. This is exactly what this paper does, specifically focusing on the domain of medical data-sharing systems.

Medical data are very sensitive data and are subject to exceptional protection measures[2]. Even though transparency is not meant to provide such measures, it allows the patients to verify that the system is taking or has taken the necessary precautions to protect their privacy.

Metrics for transparency in the medical domain have been studied in the past [19]. Those metrics are tailored to measure aspects of the information and mechanisms provided to endow a given system with transparency. However, the number of metrics proposed is heavily uneven: out of a total of eight metrics, only two were proposed for mechanisms, with one being shared with information. This might be due to fact that the methodology used to compose the set of metrics was not tailored to transparency and oversaw some aspects of it.

This is the motivation for this work. It aims at further investigating the problem of modelling transparency and defining metrics. We do so by adopting a methodology presented in [6], which proposes a meta-model to define metrics for NFRs in cloud computing. In here we slightly adapted this model to make it better represent the peculiarities of transparency in medical systems. This work completes our previous research [19]: here we propose a Model-Driven Engineering (MDE) representation of the requirements, and on that representation we clarify and extend the metrics for transparency.

The remaining of this work is structured as follows: Section 2 presents the literature related to this work. In Section 3 the details about transparency, its requirements and metrics are described. In Section 4 the high-level components of the model for transparency are explained, while Section 5 and Section 6 present the components focusing in the sub-properties of transparency. Finally, Section 7 concludes this work and presents future research directions.

[1] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation). See in particular Article 5.1(a).

[2] *Ibid.*, Art. 9.

2 Related Works

Transparency is a multi-faceted non-functional property. In the context of medical systems it is defined in terms of *availability*, *auditability*, *accountability* and *verifiability* [20]. Availability is discussed as a property that ensures the users are able to obtain and use information related to their personal data whenever needed. Auditability is the property that allows users to verify what happened to their personal data. Accountability enables users to monitor the usage of their data and hold a person accountable in case of misuse. Auditability and accountability are related to verifiability: the first is equivalent to it when applied to personal data, while the second is a more specific interpretation of verifiability.

The multi-faceted structure that characterizes transparency poses challenges to model and measure it. Conventional modelling and measuring approaches are not suited to represent its peculiarities, and methodologies focused on transparency, to the best of our knowledge, only cover part of the problem. A model for transparency, for example, has been proposed in [12]. The authors propose a model for representing transparency requirements and their relationship with other requirements, and extend an existing methodology to encompass the identification and validation of those requirements. Even though this work presents a valid approach for modelling transparency requirements, it does not consider the problem of measuring the quality of its implementation.

However, there are other relevant works on the properties composing transparency. Accountability, for example, has been systematically analysed [6] proposing a UML meta-model for defining metrics. This model helps reasoning on complex properties, structuring them into more basic ones whose metrics are simpler to define. Though designed with the purpose of measuring accountability, the model is not strictly tailored for it and can be generalised for other properties.

There are other works proposing approaches to express requirements based on MDE techniques (e.g., [2,5]). However, with respect to the previously-mentioned work on accountability, these meta-models seemed less fit to model transparency.

Metrics for transparency have also been studied from the point of view of RE. In [19] metrics are proposed to measure the quality of a transparent implementation. That work defines the qualities a transparent system should present: *informativeness*, *understandability*, *accessibility* and *validity*. Those qualities are later decomposed into eight sub-qualities, each associated with a metric to measure it. The metrics produce normalised results (ranging from 0 to 1, where 0 is the worst grade and 1 is the best). The interpretation proposed by [19] is similar to the benchmarking strategy, in which each metric is an indicator of the degree to which the properties that compose transparency are present. The metrics indicate the possibility for improvement of each property and, therefore, guide a better implementation of transparent systems.

Here, by reviewing the meta-model proposed in [6] for transparency we validate those previous metrics and we enrich them giving a more solid, model-driven, justification.

3 On transparency requirements and metrics

As mentioned in Section 2, a previous work [20] classifies transparency according to RE techniques. The classification operates under two separate viewpoints. On one side, a macroscopic classification defines the relationship between transparency and other related properties. The diagram in Figure 1 shows a visual description of this classification, pointing out how transparency is composed by the other properties (displayed as blocks).

On the other side, transparency and its sub-properties were partitioned identifying the essential requirements, extending upon previous literature [4]. These properties have been decomposed into a total of thirty-six technical requirements [20] that indicate what a medical system should do to be deemed transparent. Each of these requirements encourages the system governance to share relevant bits of knowledge about the stored data with the respective users.

This can be accomplished either by providing the users with general *information* (all kinds of "communication or reception of knowledge or intelligence"[3]) regarding the usage of data, policies and regulations, the system practices, or extraordinary events; or by offering the users *mechanisms*, that are instruments with which the user can perform operations on the data stored in the system, such as filter, select, digest, or process them, in support to some conclusion that he or she intends to take. Figure 1 outlines how the previously-mentioned properties relates to information (area in blue) and mechanisms (area in green).

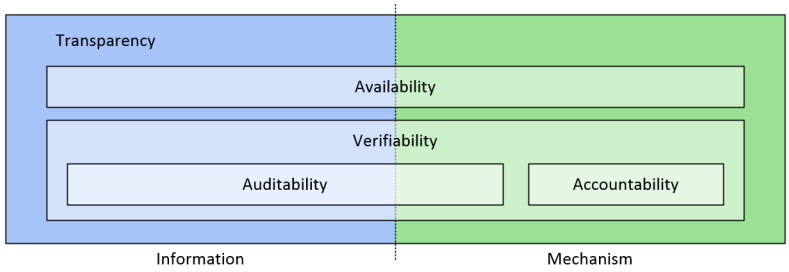

Fig. 1. Transparency and its relations with other properties.

To measure the compliance with these requirements, [19] defines eight metrics, each relating to a specific quality of the properties: *accuracy, currentness, conciseness, detailing, readability, portability, reachability*[4] and *effectiveness*. The metrics apply according to the features of each requirement. Those requirements that mandate the system to inform the user about its practices are easier to measure, and share the first seven metrics. Requirements that mandate

[3] Definition extracted from the Merriam-Webster Dictionary.

[4] Reachability was originally presented as *availability* in [19]. Here it has been renamed to avoid confusion with the transparency sub-property availability.

the system to provide a mechanism to the users are measured only by the last two metrics, with *reachability* being shared among the two families of requirements.

4 Modelling transparency properties and metrics

For the sake of simplicity, transparency is discussed and modelled with regard to five requirements, specifically the ones shown in the excerpt contained in Table 1. Even though this set may seem small, the requirements were carefully selected. Examples of each type of information-based and mechanism-based requirements, and the relevant characteristic for selecting metrics, are shown. The first three requirements demonstrate how to attain to the transparency sub-property of *availability* by providing *information* (AV.1-3), while the last two entail a *mechanism* for the transparency sub-properties of *auditability* (AU.1) and *accountability* (AC.1) respectively.

Table 1. Excerpt from the transparency requirements.

ID	**Description**
AV.1	The system must inform the user about who is responsible for handling owned data.
AV.2	The system must inform the user on how to protect data or how data are protected.
AV.3	The system must notify the user in case the policy is overridden (break the glass).
AU.1	The system must provide the user with audit mechanisms.
AC.1	The system must provide the user with accountability mechanisms.

Hereby two models explore and define the metrics for transparency and its requirements. These models also help understand the relationship between the different elements and facets of transparency (see Figure 2 and Figure 3). Details that are not immediately needed to model and validate transparency and its metrics, the goal of this work, are not shown.

The original meta-model [6] presents several elements: *Property*, *BaseProperty* and *CompoundProperty*, the objects of study; *Goal*, the high-level description of the property; *Entity*, responsible for realising the property; *Action*, representing what is executed by, or has an effect on, the entity; *Evidence* and *EvidenceProcessing*, the tangible and observable elements of the property; *Criterion*, a constraint to what should be measured; and *Metric*, *BaseMetric* and *CompoundMetric*, the methods that measures the property.

In the meta-model, the process responsible for collecting and processing the evidence is as important as the evidence itself. Hence, another evidence is proposed in association with each process to explain how it works. We slightly adapt this in our work. We accept a requirement may be implemented in several different ways, depending on the business model of each system. However,

regardless of the actual implementation, the bottom line is that the user must be able to observe the evidence in the system. This approach is alike to the software engineering technique of black-box testing, whose purpose is to test a piece of software in the absence of any knowledge of its internal structure, and based solely on the observation of its inputs and outputs [3]. Since this approach was not completely aligned to what is defined in the original meta-model [6], we slightly adapted it. This is referred throughout this paper as the "black-box approach" and is further explained in Section 5 and Section 6.

In the following we present how the elements proposed in the original meta-model are interpreted and adapted to transparency in medical systems. We first present the elements common to the two models (elements in yellow), and later, in Section 5 and Section 6, we present the remaining ones (elements in blue).

Properties The central component of these models is the *Property*, which represents what is being described. We put *Transparency* as the central property. Its composing sub-properties *Availability*, *Auditability*, *Accountability* and *Verifiability*, are represented by elements inheriting from *BaseProperty*.

In addition to transparency, two other properties are presented in this model: *Privacy* and *Usability*. They have been introduced as secondary properties that influence and need to be considered in order to provide a fair transparency. However, even in a condition of very low privacy and usability, the system may not fail to be transparent. The two properties are, therefore, not analysed in the perspective of defining metrics. This viewpoint is in accordance to what has been proposed in the literature [20], which provides more details about the relationship between those properties and transparency.

Entity and Action The properties are realised by the *Entity* element. An entity also performs or is affected by an *Action* that happens over a period of time. As transparency aims at sharing knowledge with users about how a system processes their personal data, and the system is managed by a data controller, *DataController* and *Processing* components are used.

Goal The *Goal* is the component that provides a high-level description of the purpose for which the property is being modelled, and contains a reference to the stakeholder for which the goal is oriented. The proposed model does not adopt a goal to leave the possibility of exploring all the possible facets of transparency. This element might be defined at a later stage, when applying the model to a specific scenario, in order to select the transparency requirements deemed relevant to achieve a given goal, and the metrics suited to measure them.

Criterion Any constraint that may refine the aspects of the property that should be measured is modelled by the *Criterion* component. This includes regulations, guides, stakeholders' preferences and the alike. As the current work is focused on the requirements for transparency, they were composed considering the regulations and standards for data protection, and especially those relevant to medical systems. Any other constrain are not regarded in the model as they

are beyond the scope of this work. This component is associated with the goal and, therefore, should be modelled together with it when needed.

5 Availability

The model presented in Figure 2 describes the three requirements of availability (AV.1-3). These requirements are all information-based and were selected in a way to represent every possible evidence. Because of that, Figure 2 also hints how a model of any other information-based requirement would look like.

5.1 Evidences and Evidence processing

The characteristics of each requirement are represented by the *Evidence*. It captures the elements a user can observe with respect to the property of interest. Transparency is a high-level concept, difficult to observe and measure. However, whenever the requirements for transparency are properly implemented (and

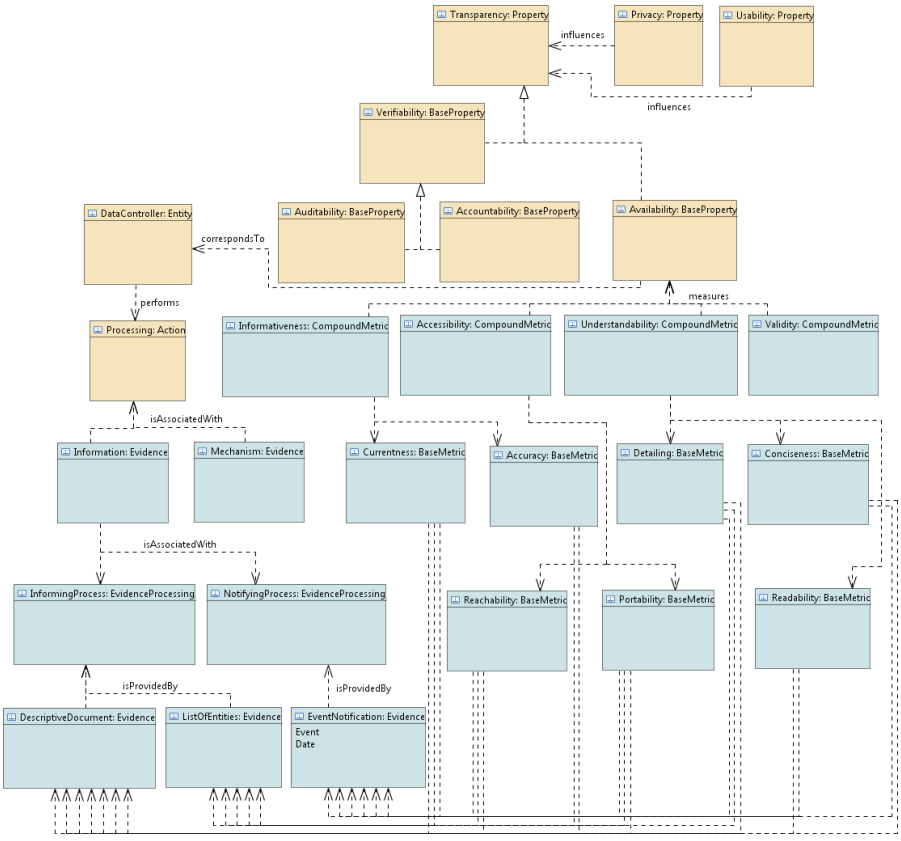

Fig. 2. Model of availability requirements.

therefore transparency is as well), the users must have access to pieces of information regarding the processing of their personal data. In other words, a sufficient amount of information provided to the user is a concrete indication that a system is actually transparent. Consequently, in this model, the *Information* component is the evidence associated with the *Processing* action.

The *EvidenceProcessing* component helps modelling the fact that the evidence, although associated with the action performed by the entity, is not produced by it. In the current model, *Information* is evidence of the fact that the *Action* of data processing is undergoing, but it is generated by other processes, which are solely responsible for informing and notifying the user. These processes are represented by the *InformingProcess* and *NotifyingProcess* elements.

According to the black-box approach described in Section 4, the analysis is centred on the evidence itself, rather than the process that collects the evidence. For example, in the requirement AV.1, "The system must inform the user about who is responsible for handling owned data", the important aspect is that users are informed about the person responsible for handling their data. It does not matter if the data controller displays a list highlighting each responsible in the system, or if it sends to the users an e-mail with the name of the person in charge of his or her data. Therefore, the focus of the analysis is on how well this information is able to satisfy the requirement. The second association between the *EvidenceProcessing* and *Evidence* elements emphasises this, it better describes what type of information is provided by each process.

Requirement AV.1 is about providing information to the users. A simple list of the people responsible for data processing should be enough for this requirement to be fulfilled. Requirement AV.2 demands that the user be informed about the protection of data. It is impossible to abstractly specify how this information looks like, but in any case it needs to describe the policies of the data controller, so it will be in the format of a descriptive document. Finally, requirement AV.3 asks for notification whenever an extraordinary event (e.g., "break the glass") happens. As it does not specify any further details, a simple notification about the occurrence of the event and the date when it happened is enough to fulfil this requirement. The *ListOfEntities*, *DescriptiveDocument* and *EventNotification* components represent the evidences in requirements AV.1-3.

5.2 Metrics

The original model classifies metrics into two types: *CompoundMetric* and *BaseMetric*. The first models metrics that are defined in terms of other metrics, while the second actually uses the evidences for the calculations. When measuring the quality of transparency implementations, there are four factors that need to be taken into account: *Informativeness*, *Understandability*, *Accessibility* and *Validity* [19]. The four factors are represented as compound metrics in this work. Whenever the data controller declares to have provided some kind of information to the users, that information is expected to have the following features: 1. to convey the precise knowledge (informativeness); 2. a comprehensible meaning (understandability); 3. the users must be able to easily obtain it (accessibility).

Validity is a quality that only concerns mechanisms for transparency, and it is about their precision and correctness of their results. Validity is presented in this model to depict the entire scenario, but will be better explored in Section 6.

Previous works [19] have defined seven metrics (represented in this model) to measure the quality of information-based requirements. *Accuracy* and *Currentness* are related to the informativeness of the evidence. *Conciseness*, *Detailing* and *Readability* concern the understandability of the evidence. Finally, *Reachability* and *Portability* refer to the accessibility of the evidence. These metrics, and a short description for each, are summarised in Table 2.

Table 2. Metrics for information-based requirements.

Metric	**Compound**	**Description**
Accuracy	Informativ.	How much the information matches the real process of the system.
Currentness	Informativ.	How timely is the information.
Conciseness	Understand.	How straightforward is the information.
Detailing	Understand.	Whether the information is detailed enough for the general understanding of its subject.
Readability	Understand.	How easy it is for a user to read and understand a text.
Reachability	Accessibility	How easy it is for a user to reach the information.
Portability	Accessibility	How easy it is to transfer and use an information in different systems.

By highlighting the specific pieces of evidence that are used to model transparency and its sub-properties, it is possible to refine the metrics and define which ones are suitable to be applied to each type of information. In particular, *Accuracy*, as a metric intended to compare statements about the data controller process and intentions to its actual practice, is not suitable for measuring events notification. That is because the events are considered as extraordinary occurrences, such as overriding an access control policy, or a security breach. Since they are unexpected, the user might not find any further information apart from the mere notification to compare them against. *Conciseness* and *Readability* are also not suitable for application to all kinds of information. The reason is that these metrics operate on a piece of information in the form of a text made up of sentences. As such, evidence in other forms, e.g., a list, might not be evaluated using those metrics. Apart from these, the other metrics remain applicable to all kinds of information-based evidences.

6 Auditability and Accountability

The model in Figure 3 describes the requirements for Auditability and Accountability (AU.1 and AC.1 respectively). These requirements are highly representative for the mechanism-based family: all evidence is represented therein and,

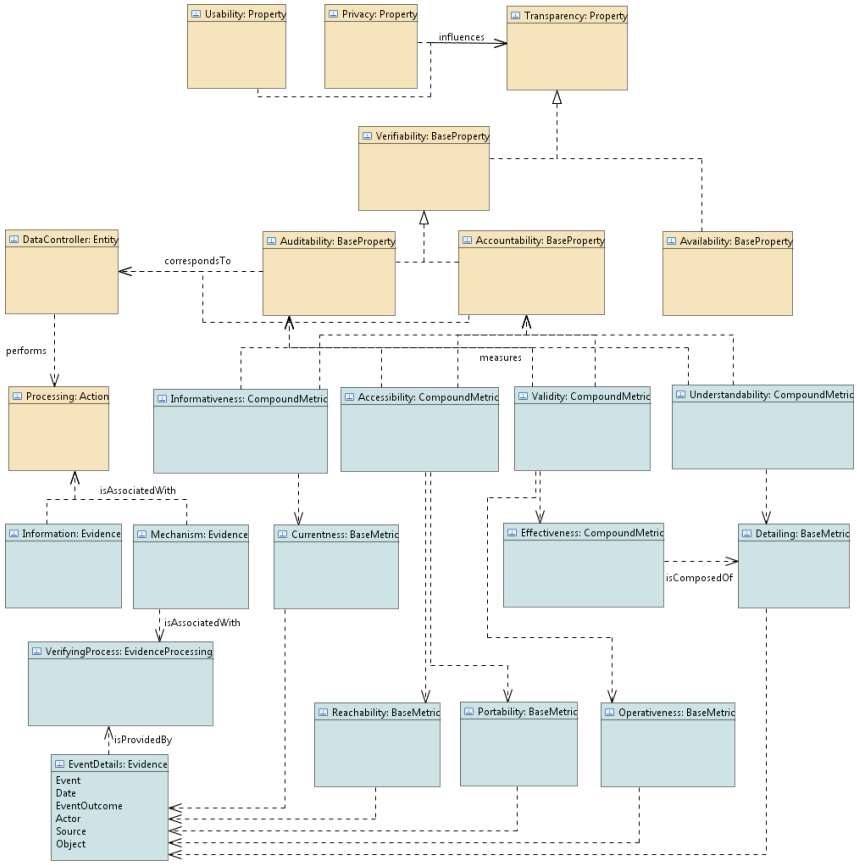

Fig. 3. Model of auditability and accountability requirements.

we think, any other mechanism-based requirements can be modelled in a similar way. Since the model overlaps with that depicted in Section 5 we will discuss only the part that is new.

6.1 Evidences and Evidence processing

Using an argument similar to that we used in Section 5, a system that complies with AU.1 and AC.1 must give users access to some sort of mechanism to verify how their data have been processed. In the model, this is represented by the evidence component *Mechanism*, and by the associated evidence processing component *VerifyingProcess*.

In the domain of medical systems, auditability and accountability are commonly interpreted as properties about access control (e.g., [7, 10, 22]). As such, they allow the users to monitor how and by whom their data has been accessed,

used, and modified[5]. The concepts of "access", "usage" and "modification" are interpreted in this work as the basic actions for persistent storage: CRUD (create, read, update and delete). In the following, auditability and accountability will be regarded as mechanisms with respect to those actions.

As each requirement may be implemented in several different ways, depending on the business model of the system implementing it, the analysis is based on the evidence they produce (as explained in Section 4). The question, then, is how the evidence of auditability and accountability mechanisms should be structured.

RFC 3881 [11] defines the format and minimum attributes that need to be captured in order to provide auditability and accountability for health systems. This document describes the data to be collected for four different events, among which patient's data events. It states that the system should document "what was done, by whom, using which resources, from what access points, and to whose medical data". On this basis, the mechanism's output should contain the following event details: 1. event identification: description of the action performed; 2. date and time; 3. event outcome: whether it was successful or not; 4. actor identification: who performed the event; 5. source: details from where the event was performed (user interface, application, etc.); 6. object identification: the data that suffered the actions. These details are abstracted in our model by the evidence component *EventDetails*.

6.2 Metrics

In the previous literature [20], only two metrics were defined to measure mechanisms-based requirements: *Effectiveness* (how satisfactory the mechanisms is by checking whether or not its goal has been reached) and *Reachability* (how easy it is for the users to reach the mechanism). The new perspective of this work resulted in the definition of one new metric and the redefinition of four other. Table 3 summarises the result of this analysis. Apart from the new metric defined, the other four are not going to be thoroughly discussed here, as they were analysed in due detail in a previous work [20].

Table 3. Metrics for mechanism-based requirements.

Metric	**Compound**	**Description**
Reachability	Accessibility	How easy it is for a user to reach the mechanism.
Portability	Accessibility	How easy it is to transfer/use the mechanism output in different systems.
Currentness	Informativ.	How up-to-date is the mechanism.
Effectiveness	Validity	How satisfactory is the mechanism provided.
Operativeness	Validity	Whether the mechanism functions and produces an appropriate effect.

[5] The interpretation adopted here also seems to be the one followed by ISO/TS 18308:2004. See in particular section 5.4.6. [9].

Reachability This metric has been defined in the previous work [20] as a function of the number of interaction necessary to reach the mechanism, but no strategy was presented to help the evaluator identify a mechanism. That was a weakness as many possible implementation are acceptable. In this work, a mechanism is considered in place whenever users can reach its output, even if a tool or plug-in is not visible to them. As such, the reachability of the mechanism has been redefined. It is measured with the same function as the previous metric, but instead it considers the number of interactions until its output is reached. The metric is represented in the model by the *Reachability* base metric.

Portability and Currentness The same reasoning applies to portability and currentness. They were originally conceived to measure the quality of information, but with the new black-box approach it is possible to apply them to the mechanism by considering its output. These metrics measure to which extent the mechanism provides information that could be easily used in other systems, and how up-to-date it is. They are represented in the model by the *Portability* and *Currentness* base metrics.

Effectiveness This metric is also being redefined here with respect to the previous work [20]. It was originally considered as a base metric that evaluated whether the mechanism was providing the expected result. By decomposing the mechanism into evidences, effectiveness can be defined in terms of the output. The new metric partly overlaps with the previously-defined *Detailing* metric. In other words, a mechanism is effective if the output it provides contains enough details to understand whether and by whom the personal data has been accessed and used (i.e., the goal of the mechanisms in requirements AU.1 and AC.1). As a consequence, in the new model *Effectiveness* is a compound metric element by the *Detailing* base metric.

Operativeness Something is said to be operative if it is functioning and "producing an appropriate effect"[6]. This metric proposes a strategy for defining whether or not a mechanism is operative. It is inspired by the black-box tests of the audit type [18] and uses the technique of equivalence partitioning [14]. It consists in partitioning the input domain of a mechanism into equivalence classes, in such a way that it is reasonable to assume that testing a value in a given class is equivalent to testing any other value in the same class. In this context, the equivalence classes are based on the actions executed in the system that a mechanism should process, and the test consists on executing these actions and observing the output to verify if they were processed by the mechanism.

For requirements AU.1 and AC.1, a reasonable set of equivalence classes can be the CRUD actions. This set of equivalence classes can be expressed as $E = C \cup R \cup U \cup D$, the union of all possible actions, where C contains create actions, R contains read actions, and so on. Measuring the operativeness metric requires to select a sub-set of actions $A = \{a_0, a_1, \ldots, a_{k-1}\} : (A \subseteq E)$, that

[6] Definition extracted from the Merriam-Webster Dictionary.

contains at least one action of each equivalence class (i.e., $(A \cap C \neq \emptyset) \wedge (A \cap R \neq \emptyset) \wedge (A \cap U \neq \emptyset) \wedge (A \cap D \neq \emptyset)$) and test it. The test consists in verifying if the actions were correctly processed and reported in the mechanism's output. If one action is not reported, or it is not possible to verify (e.g., deceptive or inconsistent information provided as output), the entire test fails. In particular, if the set of actions A contains k actions, and the number of actions that can be verified is represented by n, the result of test $\mathcal{T}_A$ can be expressed as shown in Equation (1). The result of the operativeness metric $\mathcal{O}$ is the tuple of all actions tested and the result of the test, as shown in Equation (2). This metric is represented in the model with the *Operativeness* base metric element.

$$\mathcal{T}_A = \lfloor n/k \rfloor \tag{1}$$

$$\mathcal{O} = (A, \mathcal{T}_A) \tag{2}$$

The operativeness metric presents a strategy to rationally reason about a mechanism's functioning, without delving into subjective aspects, such as whether or not the mechanism's output conveys satisfactory knowledge. This metric is conservative, meaning that it considers that one counter-example is enough to show that a mechanism is not properly functioning (this is represented by the floor function in Equation (1)). The output of 1 can be interpreted as an indication that the mechanism has performed as expected, supporting and inspiring a sense of confidence. Although, it must be noted that the operativeness of a mechanism is always measured with regard to one specific set of actions (here represented by A). Its result is therefore always accompanied by the set of actions tested, lest the result be meaningless and the test not be replicable. Each equivalence class should be measured, so it is necessary to select at least one action from each of those. If it is not possible to select one action from a particular class, it means that it is not possible to verify that class, and the test should be considered unsuccessful. The metric is flexible and allows the evaluator to decide how to couple the actions into classes, so it is possible to decide on the granularity of the test. The most suitable equivalence classes and granularity strictly depend on the peculiarities of the system implementing the mechanism. Discussing what classes can be the most appropriate for a specific system, is outside the scope of this work.

7 Discussion and Conclusions

This paper proposes a MDE approach to the transparency of a system, introducing a UML model for transparency and its metrics based on a meta-model previously used in RE. The model unfolds transparency into basic and concrete elements, much easier to measure than the high-level property of transparency itself. Through the use of this model, metrics for transparency undergo a significant uplift, with a refinement of those already defined and the introduction of new ones, and with the evidence at the centre of the analysis. By regarding

accountability as one of the building blocks for transparency, the meta-model (which was tailored to model accountability) is applicable to the domain of transparency in general.

This work contributes to the understanding of transparency. It sheds a light on how to decide if a solution is in compliance with the data protection regulation and the dramatic changes it will bring about when it finally becomes applicable, on 25 May 2018[7]; it does so by improving and extending the set of metrics that can be used to measure transparency.

Previously, the transparency analysis comprehensively embraced the information-based requirements, but it fell severely short on the mechanisms. The MDE approach presented in this paper allows to fill in that gap. The methodology clarifies that the implementation of transparency mechanisms is not so much relevant as the output they deliver to the users. Under this new perspective, it is possible to gather how well implemented a transparency mechanism is by looking at its output. Three new metrics were defined and adapted for mechanism-based requirements, and the two already-existing ones were fine-tuned.

A future research direction that stretches along the line of this work is to define a methodology to assess the trustworthiness of the transparency mechanism. Trustworthiness has several interpretations, but one that seems to be well accepted is that it is related to the assurance the system will perform as expected (e.g., [1, 13, 15]). In this sense, trustworthiness can be presented as a more complex metric that builds on top of the operativeness metric, using a well defined methodology to select the actions tested in that metric. Such a methodology, when implemented in a simple and streamlined software tool, could be independently run by any user with access to the transparency mechanisms, providing grounds to reinforce the trust framework of the system.

This work also calls for some development on the technical side. The models introduced herein could be implemented using some modelling support tool and meta-model such as Ecore [21] or some other formalism that integrates seamlessly with the formalism used in software engineering tools. By integrating transparency requirements into tools used in software engineering, such tools would allow to design and develop software services addressing transparency throughout all the software development life-cycle.

References

1. Alhadad, N., Serrano-Alvarado, P., Busnel, Y., Lamarre, P.: Trust evaluation of a system for an activity. In: Int. Conf. on Trust, Privacy and Security in Digital Business. pp. 24–36. Springer (2013)
2. Baudry, B., Nebut, C., Le Traon, Y.: Model-driven engineering for requirements analysis. In: Proc. of the 11^{th} IEEE Int. Enterprise Distributed Object Computing Conference. pp. 459–466. IEEE (2007)
3. Beizer, B.: Black-box testing: techniques for functional testing of software and systems. John Wiley and Sons (1995)

[7] Regulation (EU) 679/2016, Article 99.2.

4. Cruzes, D.S., Jaatun, M.G.: Cloud provider transparency: A view from cloud customers. In: 5^{th} Int. Conf. on Cloud Computing and Services Science. pp. 30–39 (2015)
5. Denger, C., Berry, D.M., Kamsties, E.: Higher quality requirements specifications through natural language patterns. In: Proc. of the IEEE Int. Conf. on Software: Science, Technology and Engineering. pp. 80–90. IEEE (2003)
6. Fernández-Gago, C., Nuñez, D.: Metrics for Accountability in the Cloud, Lecture Notes in Computer Science, vol. 8937, pp. 129–153. Springer International Publishing (2015)
7. Flores, A.E., Vergara, V.M.: Functionalities of open electronic health records system: A follow-up study. In: 6^{th} Int. Conf. on Biomedical Engineering and Informatics. pp. 602–607. IEEE (2013)
8. Hildebrandt, M.: Defining profiling: A new type of knowledge? Profiling the European Citizen pp. 17–45 (2008)
9. International Organization for Standardization: ISO/TS 18308:2004 Health informatics - Requirements for an electronic health record architecture (2004)
10. King, J.T., Smith, B., Williams, L.: Modifying without a trace: general audit guidelines are inadequate for open-source electronic health record audit mechanisms. In: Proc. of the 2^{nd} ACM SIGHIT Int. Health Informatics Symposium. pp. 305–314. ACM (2012)
11. Marshall, G.: RFC 3881 - Security Audit and Access Accountability Message XML Data Definitions for Healthcare Applications. Request for comments, Internet Engineering Task Force (IETF) (2004)
12. Meis, R., Heisel, M.: Computer-aided identification and validation of intervenability requirements. Information 8(1), 30 (2017)
13. Mohammadi, N.G., Heisel, M.: A Framework for Systematic Analysis and Modeling of Trustworthiness Requirements Using i* and BPMN. In: Int. Conf. on Trust and Privacy in Digital Business. pp. 3–18. Springer (2016)
14. Myers, G.J., Sandler, C., Badgett, T.: The art of software testing. John Wiley and Sons (2011)
15. Pavlidis, M., Mouratidis, H., Kalloniatis, C., Islam, S., Gritzalis, S.: Trustworthy selection of cloud providers based on security and privacy requirements: Justifying trust assumptions. In: Int. Conf. on Trust, Privacy and Security in Digital Business. pp. 185–198. Springer (2013)
16. Schwab, K., Marcus, A., Oyola, J.O., Hoffman, W., Luzi, M.: Personal data: The emergence of a new asset class (2011), `https://www.weforum.org/reports/personal-data-emergence-new-asset-class`, Last accessed in April 2017
17. Schwartz, P.M.: Property, privacy, and personal data. Harvard Law Review 117(7), 2056–2128 (2004)
18. Smith, B.: Systematizing security test case planning using functional requirements phrases. In: Proc. of the 33^{rd} Int. Conf. on Software Engineering. pp. 1136–1137. ACM (2011)
19. Spagnuelo, D., Bartolini, C., Lenzini, G.: Metrics for Transparency, pp. 3–18. Springer International Publishing, Cham (2016)
20. Spagnuelo, D., Lenzini, G.: Transparent medical data systems. Journal of Medical Systems 41(1), 8 (2016)
21. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF. Eclipse, Addison-Wesley, 2^{nd} edn. (2009)
22. Tong, Y., Sun, J., Chow, S.S., Li, P.: Cloud-assisted mobile-access of health data with privacy and auditability. IEEE Journal of Biomedical and Health informatics 18(2), 419–429 (2014)