D. Nuñez, C. Fernandez-Gago, S. Pearson, and M. Felici, "A Metamodel for Measuring Accountability Attributes in the Cloud", *2013 IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2013)*, pp. 355-362, 2013.
http://doi.org/10.1109/CloudCom.2013.53
NICS Lab. Publications: https://www.nics.uma.es/publications

A Metamodel for Measuring Accountability Attributes in the Cloud

David Nuñez, Carmen Fernandez-Gago
Network, Information and Computer Security Laboratory
Universidad de Málaga
Málaga, Spain
Email: {dnunez, mcgago}@lcc.uma.es

Siani Pearson, Massimo Felici
Security and Cloud Lab
Hewlett-Packard Laboratories
Bristol BS34 8QZ, United Kingdom
Email: {siani.pearson, massimo.felici}@hp.com

***Abstract*—Cloud governance, and in particular data governance in the cloud, relies on different technical and organizational practices and procedures, such as policy enforcement, risk management, incident management and remediation. The concept of *accountability* encompasses such practices, and is essential for enhancing security and trustworthiness in the cloud. Besides this, proper measurement of cloud services, both at a technical and governance level, is a distinctive aspect of the cloud computing model. Hence, a natural problem that arises is how to measure the impact on accountability of the procedures held in practice by organizations that participate in the cloud ecosystem. In this paper, we describe a metamodel for addressing the problem of measuring accountability properties for cloud computing, as discussed and defined by the Cloud Accountability Project (A4Cloud). The goal of this metamodel is to act as a language for describing: (i) accountability properties in terms of actions between entities, and (ii) metrics for measuring the fulfillment of such properties. It also allows the recursive decomposition of properties and metrics, from a high-level and abstract world to a tangible and measurable one. Finally, we illustrate our proposal of the metamodel by modelling the transparency property, and define some metrics for it.**

***Index Terms*—Metrics, Non-functional properties, Metamodel, Cloud computing, Accountability**

I. Introduction

According to the NIST definition of cloud computing [1], the cloud model is composed of five essential characeristics, namely, on-demand self-service, broad network access, resource pooling, rapid elasticity and measured service. The latter characteristic, which is the scope of this paper, is defined by the capacity of cloud systems for measuring aspects related to the utilization of services, in order to provide automatic control and optimization of the usage of cloud resources, and ultimately, to support transparency and enhance trust of cloud consumers with regard to cloud providers.

Metrics in cloud computing are also of paramount importance for other reasons. For instance, metrics can also be derived on the consumer side, enabling cloud consumers to monitor the quality of service of the cloud provider and to verify the compliance of agreed terms. Metrics are also a tool that facilitate the decision making process of cloud consumer organizations, as they can be used for making informed decisions with regard to the election and evaluation of cloud providers.

As for cloud service governance, metrics are very useful means for assessing performance of operational processes and for demonstrating the implementation of appropriate practices through the provision of quantifiable evidence of the application of such practices. In other contexts, such as quality assurance, software development or project management, metrics are traditionally used as a tool for monitoring progress, assessing compliance, and facilitating and refining the decision-making process within an organisation.

From a broader perspective, metrics are a key aspect of organizational maturity models because of their role in quality assessment, monitoring of processes performance and support of management decisions, and will be highly relevant when addressing the definition of cloud maturity models [2]. In theory, a mature organisation (from the perspective of maturity models) should present a quantitative, and hence, measurable behaviour. Mature organisations are therefore characterized by an ingrained use of metrics within their internal processes. The adoption and systematic use of metrics is an indispensable practice for organisations that strive to achieve a repeatable and optimizing behaviour. For all these reasons, it is clear that the definition and usage of meaningful metrics is a crucial requirement of the cloud computing model.

Cloud governance, in particular data governance in the cloud, relies on a variety of technical and organizational means, such as policy enforcement, risk management, incident management and remediation. Accountability is a high-level concept that entails all these practices. In general, accountability deals with being able to demonstrate that the accounts provided by an organisation (to regulators, auditors, data subjects or other service providers) are adequate and appropriate for the context, and implementing mechanisms for responding to the situation (including sanctions and remediation) if this is not the case. Since measuring is a distinctive facet of the cloud model, it is a natural question to wonder how to effectively perform measurements of the fulfillment of such requirements.

In this paper, we focus on measuring accountability of cloud services. We propose a metamodel that permits to model non-functional properties of cloud services, and in particular, those that influence accountability. These properties are identified and discussed by the Cloud Accountability Project (A4Cloud) [3], and referred as *attributes of accountability*. Those include transparency, verifiability, observability, liability, responsability, attributability and remediation (for details see [4]). Our goal is to use this metamodel as part of a methodology for elicitating properties and defining metrics for them.

The rest of this paper is organized as follows. In Section II, we identify the problems that we face when addressing the definition of metrics for accountability attributes. In Section III, we describe the proposed metamodel and explain in detail its elements. Section IV shows how this metamodel can be used for modelling properties of accountability; in particular, the Transparency property. In Section V, we review the related work that is relevant to our proposal, such as similar metamodels and research on the definition and assessment of non-functional properties. Finally, Section VI concludes the paper and outlines the future work.

II. Justification of the Metamodel

The A4Cloud Project has identified several attributes that are relevant for accountability. Within this project, the accountability concept is decomposed into several properties that influence or are influenced by it. One of the goals of this project is to come up with meaningful measurement techniques for such kind of properties of accountability, in order to assess the level of accountability of service providers in the cloud.

These properties belong to the family of *non-functional properties* (also referred to as *attributes*), which include all properties that are not directly related to functionality, but to a quality or behavioural attribute of a system [5]. Non-functional properties, such as the ones related to security and privacy, are of key importance with regard to the analysis and evaluation of the different aspects of a system, a service or an organization, such as quality and trustworthiness. However, their evaluation is traditionally complicated because of several reasons. Firstly, because of their subjective and ambiguous nature; secondly, non-functional properties usually present multi-dimensionality, possessing several facets; and finally, in some cases, the optimization of a non-functional property may be inconsistent with others.

The goal of defining meaningful measures for accountability attributes is subject to the problems associated with non-functional properties. We currently lack methodologies and tools for properly defining and evaluating such properties. As aforementioned, one of the main problems of this kind of properties is their lack of a clear definition, as they are usually described in abstract terms that are not useful from a measurement perspective. For this reason, sometimes it is very difficult to assess if such a property has been met since there is no clear-cut criteria for that.

Taking *transparency* as an example of a non-functional property, the Cloud Industry Forum's Code of Practice [6] broadly speaking interprets transparency in the sense of transparency between the data processor and the data controller. However, within the data protection community, transparency instead usually refers to transparency of the data controller with respect to the data subject. This kind of inconsistency causes difficulties during the process of defining metrics.

It is therefore important to pay special attention to these issues during the elicitation of non-functional properties that a system or organization must meet. In particular, we identify the following problems that arise at a semantic level:

- Level of abstraction: Most of the time, non-functional properties are defined in a very abstract fashion, which makes them of little use from the metrics point of view. Another problem is the variety of levels of abstraction between properties.
- Ambiguity: Natural language permits vague definitions, prone to different interpretations. Definitions also tend to be similar among some properties, which facilitate their overlapping. We identify two problems:
  - *Homonymy*: The same name is used to designate different properties, as in the case of transparency.
  - *Synonymy*: A property is designated by different names. This could be a desired effect, as each name could identify a subtle variation of the property; however, in reality, most of the time, designations are arbitrarily interchanged.
- Subjectivity: Non-functional properties are often interpreted differently depending on the stakeholder and are very sensitive to the context of application, so in most cases there is no widely accepted definition for this kind of properties.
- Overlapping of properties: In most cases, some of the identified properties partially or fully overlap with others. This is not negative by itself, as it is natural that two properties share some characteristics; however, from the metrics point of view, this phenomenon leads to confusion. Clearer and more disjunct definitions are needed.
- Interdependencies between properties: An exhaustive analysis of property interlinks would probably have as a result an intricate network of influences and dependencies between properties. This also makes the process of properly specifying properties and defining measurement techniques for them very difficult.

Most of these problems arise from the use of natural language for the definition of these properties. Hence, a more formal approach is needed for modelling the different concepts related to the measurement of accountability properties.

We propose a model-driven approach that includes the definition of a metamodel for describing metrics and accountability properties. The goal of this metamodel is to serve as a language for describing: (i) accountability properties in terms of entities, evidence and actions, and (ii) metrics for measuring them. Note that this metamodel could be extended for its application to non-functional properties in general, however, this is out of the scope of this paper since we are currently focused on those related to the accountability concept.

One of the main aspects of this metamodel is that metrics are defined to take two main kinds of inputs: *Evidence* and *Criteria*. From our point of view, any assessment or evaluation (i.e, a metric) can only be made using as input some tangible and empirical evidence, such as an observation, a system log, a certification asserted by a trusted party, a textual description of a procedure, etc. That is, a metric does not directly measure a property of a process, a behaviour, or a system, but uses the evidence associated with them in order to derive a meaningful measure. That is the idea that we are trying to capture in our model: Evidence is the fundamental support of any evaluation method and is what gives an objective dimension to assessments. On the other hand, criteria are all the elements that convey contextual input that may constrain what should be measured, such as stakeholder's preferences, regulations and policies. It is clear then that each metric will have different nature depending on the criteria. Therefore, in our model, both *Evidence* and *Criteria* are central to the definition of metrics.

III. A Metamodel for Metrics for Accountability Properties

In this section, we will present our metamodel (see Figure 1), and provide a detailed description of each of its elements and the relations among them.

- Goal: High-level description of the property (or family of properties) that is modeled. These elements also contain a reference to the stakeholder (or stakeholders) for which the goal is oriented.
- Property: As mentioned earlier, non-functional properties are qualities or behavioural characteristics of an entity. Ideally, properties can be distinguished quantitatively or qualitatively by some evaluation method; however, properties may be defined as very high-level concepts. Thus, we consider that properties can be further decomposed into more basic ones in some cases. In these cases, BaseProperty elements can be defined in terms of entities and the actions between them, whereas CompoundProperty elements are defined in terms of other properties, making possible a top-down decomposition of properties, from a high-level and abstract way to a tangible and more accessible one. CompoundProperty elements then have a connective attribute, which is used for describing the logical connective used for combining properties. In addition, properties may also influence other properties, not necessarily taking part of a composition relation; the model then permits to express these influence relations between properties.
- Entity: This element is used to describe the entity that meets the modeled property. An entity is a physical or conceptual object that performs actions and that meets properties. For example, an organization, a process or a system can be considered as entities.
- Action: We define this as a process that occurs over a period of time and is performed by or has an effect on entities. Even though, actions have an effect in the environment, we cannot deal directly with these consequences, but with the evidence associated to them.
- Evidence: We define evidence as a collection of information with tangible representation about the effect of actions. Evidence is used to support a metric. That is, evidence is not an abstract concept about the consequence of activities, but actual data that can even be processed by a machine. However, evidence may come from sources with different levels of certainty and validity, depending on the method of collection or generation.
- EvidenceProcessing: In our model, we assume that evidence, although it is associated to the effect of actions, does not directly stem from them. Instead, evidence is originated or collected by means of an EvidenceProcessing element. In this way, we model the fact that there may not exist a perfect correlation between the effects or consequences of actions and the evidence associated with them. The EvidenceProcessing element makes this difference explicit. With the inclusion of this element in our metamodel, we emphasise that the method of collection and processing of evidence is as important as the evidence itself. For this reason, there should also be evidence associated with each EvidenceProcessing element, describing how it works. Such evidence may be used by a metric during the evaluation process.
- Metric: We define this as an evaluation method for assessing the level of satisfaction of a non-functional property in a quantitative or qualitative way, on the basis of evidence and contextual criteria. Metrics can be of two types: BaseMetric for metrics that use evidence as inputs for their calculations, and CompoundMetric for aggregated metrics that are defined as a function of other metrics. Aggregated metrics may rely on auxiliary metrics that are not associated with any property and that are defined solely for facilitating the definition of the parent metric. In both cases, metrics may use Criterion elements for guiding the evaluation with respect to the context of the metric. This element has the following fields:

  - Scale: This field describes the type of measurement scale used in this metric. The scale can be either *nominal*, *ordinal*, *interval* or *ratio*. More details are given in Section III-A. Nominal and ordinal metrics are often considered as *qualitative* metrics, whereas interval and ratio metrics are *quantitative*.
  - Unit: This field represents the measurement unit adopted as standard for measuring the property. The definition of a measurement unit is only necessary in the case of quantitative metrics.
  - Constraints: This field conveys the contextual constraints that may affect the application and validity of the metric.

- Criterion: This element captures all the contextual input that may constrain what should be measured by the metric, such as regulation, best practices, organisational policies and contracts, and stakeholders' preferences. It could be the case that one could define different metrics for the same property. The assessment methodology for each metric will depend on the contextual input given for the metrics evaluation. The Criterion element will be the responsible of conveying such contextual information.

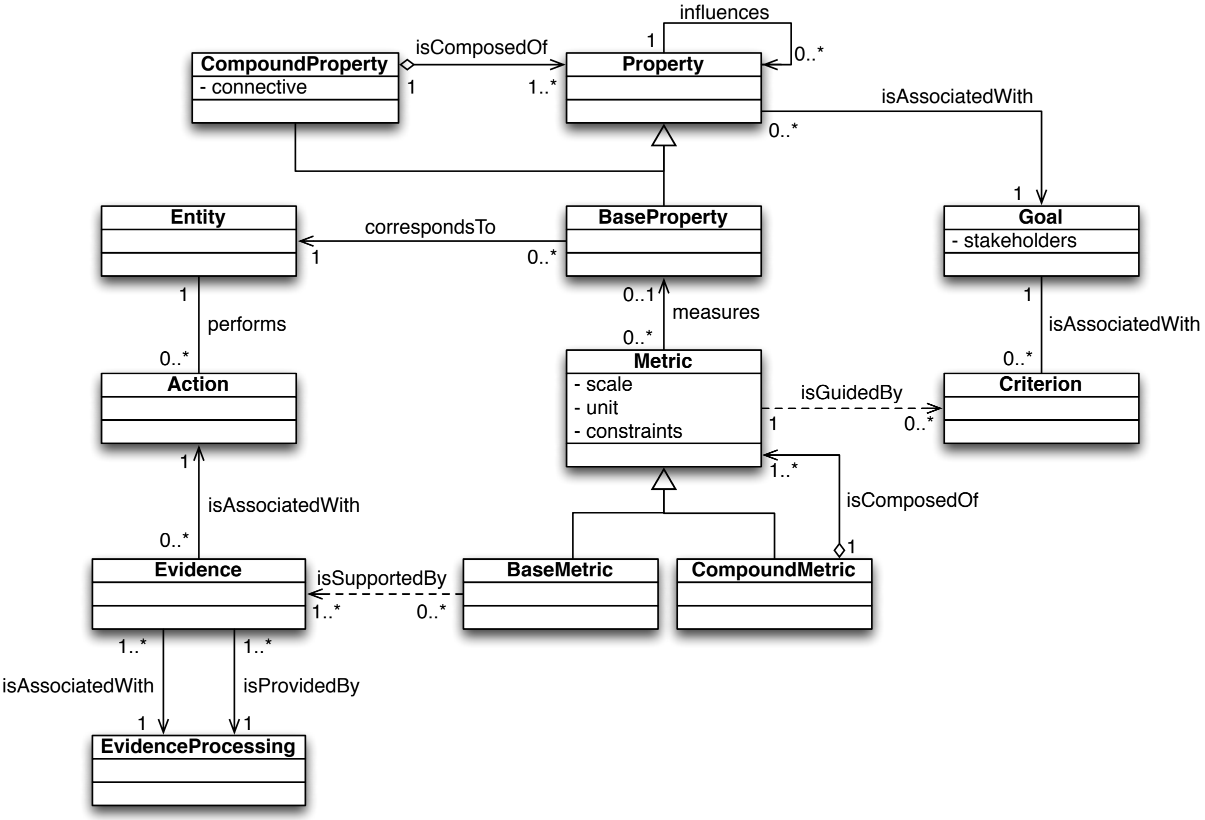


Fig. 1. Metamodel for Metrics for Accountability Attributes

The intention behind this metamodel is to be used as part of the process of elicitation and evaluation of accountability properties in a cloud context. Hence, the stakeholder who is interested in assessing such properties would be the one that takes the role of owner of the model described using this metamodel. Each particular model defined using this language reflects the viewpoint of the model owner with regard to the context of application. Customization of models to specific situations is then done in different ways:

- Decomposition and interlinking of properties: the modeler can freely identify the goals and their associated properties, which can be further decomposed into other subproperties or interlinked through influence relations.
- Modeling of entities and their actions: Entities and actions can be modeled with the level of abstraction desired by the model owner, as the metamodel simply dictates that entities perform actions.
- Identification of meaningful evidence sources: the EvidenceProcessing element is used to model the sources of evidence that stem from the effect of actions.
- Definition of different metrics in terms of evidence and criteria: the possibility of defining different metrics for the same property is another characteristic that supports the customisation of models. Thus, the context and preferences of the model owner with regard to evaluation of properties can be reflected. Each metric would have different sources of evidence and criteria.

A. Scales of Measurement

In the classical theory of measurement [7], the scales of measurement (or levels of measurement) are a set of categories for classifying measurement methods regarding their characteristics. Identifying the scale for each particular metric is essential for interpreting and analysing its results. Moreover, since each scale has a set of permitted operations, knowing its scale allows us to assess the validity of a compound metric.

- Nominal scales: This type of scale is applicable for mapping measured properties to names or categories. It is also known as a categorical scale. Values in a nominal scale do not have any kind of relation to each other. For this reason, only the equality operation (=) is permitted for nominal values. From a statistical viewpoint, only modes can be computed.
- Ordinal scales: This scale permits assigning an order relation to its values, which is used to put measurements in order. For this reason, ordinal scales are said to have magnitude. However, there is no information for measuring the differences between values. A simple example of this scale is the set of values “Low – Medium – High”. There is an order relation that permits to state that High is greater than Medium, which in turn is greater than

Low, but it makes no sense to measure the difference between Low and Medium. Ordinal scales are also nominal and permit using equality (=) and inequality ($\leq$) operations, as well as medians and percentiles. Certain non-parametric statistical tests that only require ordinal data, known as ranking tests [8], can also be performed.
- Interval scales: This type of scale permits measuring differences or distances between values. Additionally, interval scales are also ordinal scales. Thus, their values can be compared and ordered. Interval scales permit additions and substractions of their values. Therefore, means and standard deviations can also be computed. However, multiplications and divisions, and hence any other operations that depend on those, such as ratios, cannot be performed.
- Ratio scales: This type of scale improves interval scales by adding a meaningful zero value (interval scales can have zero values but its placement is arbitrary). Ratio scales are also interval scales, so all the operations that are valid for interval scales apply here. In addition, multiplication and division are also meaningful. Nominal and ordinal metrics are often grouped as qualitative metrics, whereas interval and ratio metrics are quantitative.

It is important to have these concepts clear when defining metrics, since it is usual to create metrics that are not valid from the point of view of measurement. In particular, when a measurement is done upon a qualitative attribute, one cannot use an interval or ratio scale without a proper justification. An example of this is given at the end of Section IV.

IV. Modelling Attributes for Accountability

This metamodel is devised within the framework of the A4Cloud project, which aims to address data governance in cloud computing by devising methods and tools, through which organisations can be made accountable for the privacy and confidentiality of information held in the cloud. These methods and tools combine risk analysis, policy enforcement, monitoring and compliance auditing.

One of the objectives of this project is to develop measurement techniques for the non-functional properties that influence or are influenced by accountability. Such properties, referred as *attributes of accountability,* include transparency, verifiability, observability, liability, responsability and attributability. Essentially, what is needed for ensuring accountability is to be able to demonstrate that the accounts provided by an organisation (to regulators, auditors, data subjects or other service providers) are adequate and appropriate for the context, and to have in place mechanisms for dealing with the situation (including sanctions and remediation) if this is not the case. From an organisational point of view the focus is on measuring whether the fundamental types of activities that an accountable organisation should undertake are in place and effective.

Hence, we focus on measuring the non-functional properties corresponding to the criteria for measuring or demonstrating accountability defined by [9], i.e. policies, executive oversight, staffing and delegation, education and awareness, ongoing risk assessment and mitigation, program risk assessment oversight and validation, event management and complaint handling, internal enforcement and redress, as well as similar lists (not exhaustive and not applicable to all organisations) such as those given by [10]. As part of this process, it must be shown that the organisational policies, risk assessment process and related decisions are appropriate for the business context, the privacy and security controls used within the organisation are appropriate for the business context and the obligations that an organisation has (in our context, coming from domestic data protection legislation and private contracts) are met throughout the service provision chain.

All these practices can be linked to different attributes for accountability; proper implementation and demonstration of such practices and procedures will influence positively the evaluation of their associated properties, and on the contrary, bad quality or lack of them will impact negatively. Ultimately, all of them will have an effect on accountability. Future work will cover these issues, as the focus of this paper is defining a metamodel for describing accountability properties and their associated metrics.

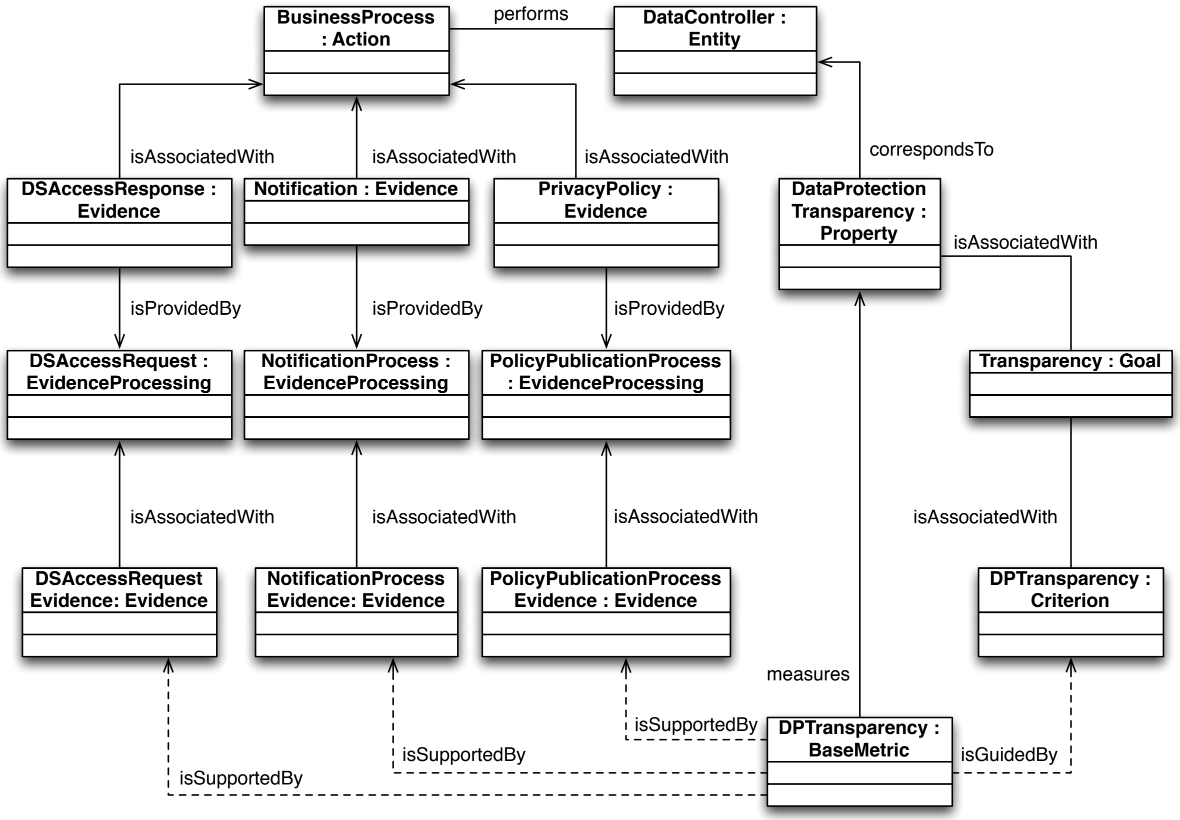
In order to illustrate our proposal, we show how one particular attribute for accountability, *Transparency*, could be (partially) modeled from one of the definitions given by the A4Cloud project [3][4]:

"[...] an accountable organisation is transparent in the sense that it makes known to relevant stakeholders the policies defined about treatment of personal and confidential data, can demonstrate how these are implemented and provides appropriate notifications in case of policy violation, as well as responding adequately to data subject access requests."

From a high-level viewpoint, a transparency metric would measure the susceptibility of an organization's policies and procedures regarding data protection to be inspected by relevant parties (such as data subjects), as well as the quality of the transparency processes held in place by the organization. Figure 2 shows a model of the Transparency attribute using our metamodel.

In this example, the high-level goal is represented by the Transparency element. This goal could have associated several properties related to Transparency. In this case, we are referring to transparency with respect to data protection (represented by DataProtectionTransparency), as we are dealing with the treatment of personal and confidential data. This property is defined upon an organization which acts as Data Controller (since it determines the purposes and means of the processing of personal data). In other words, a metric for this property would evaluate how transparent this organization (i.e., the DataController element) is with respect to data protection. In this example, the actions of the DataController are subsumed into one Action element and called BusinessProcess.

In order to achieve transparency, the DataController must implement and demonstrate the application of certain practices that contribute to enhance its transparency. Taking the above definition of Transparency into consideration, we identify three practices or transparency processes:



BusinessProcess : Action
performs
DataController : Entity
isAssociatedWith
isAssociatedWith
isAssociatedWith
correspondsTo
DSAccessResponse : Evidence
Notification : Evidence
PrivacyPolicy : Evidence
DataProtection Transparency : Property
isAssociatedWith
isProvidedBy
isProvidedBy
isProvidedBy
DSAccessRequest : EvidenceProcessing
NotificationProcess : EvidenceProcessing
PolicyPublicationProcess : EvidenceProcessing
Transparency : Goal
isAssociatedWith
isAssociatedWith
isAssociatedWith
isAssociatedWith
DSAccessRequest Evidence: Evidence
NotificationProcess Evidence: Evidence
PolicyPublicationProcess Evidence : Evidence
DPTransparency : Criterion
measures
isSupportedBy
DPTransparency : BaseMetric
isSupportedBy
isGuidedBy
isSupportedBy

Fig. 2. The Transparency property modeled using the metamodel

- Informing stakeholders about data protection policies and their implementation practices
- Notification in case of policy violation
- Responding to data subject access requests

These practices are directly mapped in our model example to the following EvidenceProcessing elements:

- PolicyPublicacionProcess. This element represents the internal procedures of the DataController towards the publication and communication of data protection policies to the relevant stakeholders. This element has associated two Evidence elements:
  - PrivacyPolicy. This Evidence is produced by the PolicyPublicationProcess; that is, the result of this process is an examinable description of the data protection policy that can be accessed by relevant stakeholders. This element by itself could not be relevant to the DataProtectionTransparency property. That is, individual privacy policies are not assessed by a Transparency metric as transparency is focused on making the policies known. In this case, only the existence of these elements could be assessed. However, the contents of the privacy policy could be interesting for measuring other properties such as Compliance of particular policies.
  - PolicyPublicationProcessEvidence. This instance of Evidence is associated with the transparency process that publishes privacy policies and describes its nature and characteristics. For example, it could answer questions such as *'Are all the policies published?' 'Are the policies consistent with the real procedures in practice?' 'Who asserts this consistency?' 'Is it self-asserted or certified by a trusted party?'*. These answers are the aspects that may influence the definition of a Transparency metric and its evaluation.
- NotificationProcess. This element represents the internal practices of the DataController with respect to notification to the relevant stakeholders about any violation of data protection policies. This element has associated two Evidence elements:
  - Notification. This element represents the Evidence generated by the NotificationProcess in case of a policy violation.
  - NotificationProcessEvidence. This element represents a description of the nature of the process of notification. That is, it answers questions such as *'Does a notification process exist?' 'Are the means of notification appropriate?' 'Are notifications consistent with privacy policies?' 'Who asserts this consistency?' 'Is it self-asserted or certified by a trusted party?'*.

- DataSubjectAccesRequestProcess. This element represents the internal procedures of the DataController for permitting data subjects to request access to their data and for properly responding to such requests. This element has associated two Evidence elements:
  - DataSubjectAccessResponse. This element is the evidence representing the response generated by the DataSubjectAccessRequestProcess in case of an access request from a data subject.
  - DataSubjectAccessRequestEvidence. This element represents a description of the characteristics of the process for permitting data subject access requests. That is, it answers questions such as *'Does a process for data subject access requests exist?' 'Is this process accessible to data subjects?' 'Is it consistent with privacy policies?' 'Who asserts this consistency?' 'Is it self-asserted or certified by a trusted party?'*.

Hence, it is the Evidence elements associated to these processes, and not the evidence produced by them, the ones that are evaluated by the DataProtectionTransparency metric. The evidence generated by these processes could be evaluated by metrics for other attributes (for example, the PrivacyPolicy evidence could be evaluated by a Compliance metric).

Based on the existence of the transparency processes that stem from the definition of Transparency (publication of policies, notification, or permitting data subject access requests), we define an example of a metric for Data Protection Transparency. This metric is based on the controls defined in a self-assessment questionnaire for cloud service providers, the CSA Consensus Assessments Initiative Questionnaire v1.1 (CAIQ) [11]; each transparency process is mapped to a specific control from the questionnaire, as shown in Table I. Moreover, we will assume that the owner of the metric gives different weights to each transparency process: 0.5 to the publication of policies, 0.3 to the notification and 0.2 to the handling of data subject access requests. Note that one may be tempted to produce a formula such as $0.5 \cdot TP_1 + 0.3 \cdot TP_2 + 0.2 \cdot TP_3$, where TP_1, TP_2 and TP_3 can take values 0 or 1 if the transparency process is implemented (respectively, publication of policies, notification, or permitting data subject access requests). Such a formula could give the impression of having defined a metric with an interval or ratio scale from 0 to 1. However, in reality there is no real meaning for the differences between the possible values. Thus a more valid approach could be to define an ordered scale such as the following, that still conveys the same intention from the owner of the metric:

- *Level 0*: No transparency processes are implemented
- *Level 1*: Only a process for handling data subject access requests is implemented
- *Level 2*: Only a process for notification is implemented
- *Level 3*: Either the process for publication of policies or the processes for notification and data subject access requests are implemented
- *Level 4*: The processes for publication of policies and data subject access requests are implemented
- *Level 5*: The processes for publication of policies and notification are implemented
- *Level 6*: All transparency processes are implemented

TABLE I
MAPPING CSA CAIQ CONTROLS TO TRANSPARENCY PROCESSES

Control Identifier	**Consensus Assessment Question**	**Transparency Process**
IS-26.1	Do you provide documentation regarding how you may utilize or access tenant data and/or metadata?	Policy publication
IS-27.1	Are systems in place to monitor for privacy breaches and notify tenants expeditiously if a privacy event may have impacted their data?	Notification
IS-26.3	Do you allow tenants to opt-out of having their data/metadata accessed via inspection technologies?	Data subject access requests

Next, we performed a real evaluation based on the responses that are publicly available at the CSA Security, Trust & Assurance Registry (STAR) [12]. We reviewed the responses from three cloud service providers and assessed their level of Data Protection Transparency according to the metric defined above. The results are shown in Table II. Note that in this example, the responses to the CAIQ controls are taken as the Evidence elements for evaluating the transparency processess.

TABLE II
EVALUATION OF REAL CLOUD SERVICE PROVIDERS FROM CSA STAR

CSP	TP_1	TP_2	TP_3	**Level**
CSP A	Yes	No	No	3
CSP B	Yes	No	Yes	4
CSP C	Yes	Yes	No	5

The definition of the metric could be more sophisticated, not just covering which transparency processes are implemented in a binary way (i.e., yes or no), but including some judgement about the degree to which they are implemented. Moreover, in this case, evidence stems from self-assessments; a more complex metric could take certification and third-party audits in consideration.

An open question is how to derive quantitative metrics from inherently qualitative attributes. From a strict point of view, one cannot simply assign a number to a quality value and perform operations. In that case, it is preferably to analyse the intended formula and produce a metric with an ordered scale, as in this example. This scale can be then more complex (i.e., with more levels) and still be valid.

Finally, it can be observed that a different definition of transparency could lead to a different model; that is the reason why we consider that a first requirement towards creating metrics is agreeing on a clear, concise and stable definition of the property to be measured, so that an appropriate model can be defined.

V. Related work

Ontologies have been used to describe non-functional properties, and could be of use to clarify the meaning of the accountability attributes. For instance, Sullivan et al. describe in [13] an ontology for several concepts related to security and trustworthiness, such as privacy, accountability, anonymity and transparency. One of the objectives behind their ontology is to serve as a basis for defining better measurability criteria. However, their results are not extensive enough, and further work that includes more concepts is needed. In [14], O'Sullivan et al. present an approach for describing non-functional properties of services using Object-Role Modelling (ORM). Although not technically an ontology (but a data model), they do provide an extensive and detailed catalogue of non-functional properties modeled using their approach. An interesting development could be to identify those that are of use within the accountability problem and to define metrics for them using our approach.

With respect to metamodels for non-functional properties, other authors have proposed different approaches. For example, De Paoli et al. present in [15] the Policy Centered Metamodel, which is used for the description of non-functional properties in the context of web services selection. However, they do not take evidence and contextual criteria as supporting elements of the evaluation of non-functional properties. Di Marco et al. introduce in [16] a property metamodel aimed to describe in a machine-readable way the non-functional properties of services. In their model, there is no hint of what elements support the evaluation of metrics. Their work does not mention either how metrics deal with different contextual criteria. There exists also similar work in the field of software quality, given its non-functional nature. For instance, Mohagheghi and Dehlen [17] propose a metamodel for specifying quality models in the context of model-driven engineering.

VI. Conclusions and Future Work

In this paper we describe a metamodel for defining metrics for accountability attributes, as part of the A4Cloud project. This metamodel is intended to be used as a modeling language for identifying the elements that form part of the definition of these type of attributes and that influence metrics for them. In particular, it permits identification of the sources of evidence that will be later used as support for performing assessments by metrics and the criteria that guides such evaluations. We also provide an example of how to use the metamodel for modeling the transparency property.

In the future we plan to apply this metamodel for describing accountability attributes, as well as the links between them in terms of decomposition and influence relations, as a first step for defining meaningful metrics for them. The next step is to identify relevant practices and procedures of organizations that participate in the cloud ecosystem and to link them to the accountability attributes. We must also define how the proposed metamodel would be used in practice to assess quantitatively specific accountability attributes, hence supporting an operational or evidence-based analysis of relationships among attributes. This would form the basis for a quantitative evaluation of accountability. Another line worth to be researched is the specification of technical mechanisms for conveying contextual criteria, which includes regulations, policies and stakeholders' preferences. At the moment, the metamodel leaves this aspect undefined. A proper language for representing this input is required for leveraging the validity of our approach. Policy languages and Natural Language Processing techniques could also be explored.

Acknowledgements

The research leading to these results has received funding from the European Union Seventh Framework Programme under grant agreement 317550, from the Spanish Ministry of Economy and Competitiveness through the SPRINT project (TIN2009-09237), and from the Junta de Andalucía through the project FISICCO (P11-TIC-07223). The first author has been funded by a FPI fellowship from the Junta de Andalucía through the project PISCIS (P10-TIC-06334).

References

[1] P. Mell and T. Grance, "The NIST definition of cloud computing," Tech. Rep. SP 800-145, 2011.

[2] S. Creese, P. Hopkins, S. Pearson, and Y. Shen, "Data protection-aware design for cloud services," in *Cloud Computing*. Springer, 2009, pp. 119–130.

[3] "The Cloud Accountability Project," http://www.a4cloud.eu/.

[4] D. Catteddu, M. Felici, G. Hogben, A. Holcroft, E. Kosta, R. Leenes, C. Millard, M. Niezen, D. Nuñez, N. Papanikolaou *et al.*, "Towards a model of accountability for cloud computing services," in *Pre-Proceedings of International Workshop on Trustworthiness, Accountability and Forensics in the Cloud (TAFC)*, 2013.

[5] N. Siegmund, "Measuring and predicting non-functional properties of customizable programs," Ph.D. dissertation, Otto-von-Guericke-Universitat, Magdeburg, Germany, 2012.

[6] Cloud Industry Forum, "Code of Practice for Cloud Service Providers," http://www.cloudindustryforum.org/code-of-practice/code-of-practice.

[7] S. Stevens, "On the theory of scales of measurement," *Science*, vol. 103, no. 2684, pp. 677–680, 1946.

[8] S. Siegel, "Nonparametric statistics," *The American Statistician*, vol. 11, no. 3, pp. 13–19, 1957.

[9] Centre for Information Policy Leadership, "Demonstrating and measuring accountability - a discussion document," 2010.

[10] Article 29 Data Protection Working Party, "Opinion 3/2010 on the principle of accountability," July 2010.

[11] Cloud Security Alliance, "Consensus Assessments Initiative Questionnaire," https://cloudsecurityalliance.org/research/cai/.

[12] ——, "Security, Trust & Assurance Registry (STAR)," https://cloudsecurityalliance.org/star/.

[13] K. Sullivan, J. Clarke, and B. P. Mulcahy, "Trust-terms ontology for defining security requirements and metrics," in *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume*. ACM, 2010, pp. 175–180.

[14] J. O'Sullivan, D. Edmond, and A. H. ter Hofstede, "Formal description of non-functional service properties," Centre for Information Technology, Queensland University of Technology, Tech. Rep., 2005.

[15] F. De Paoli, M. Palmonari, M. Comerio, and A. Maurino, "A metamodel for non-functional property descriptions of web services," in *Web Services, 2008. ICWS'08. IEEE International Conference on*. IEEE, 2008, pp. 393–400.

[16] A. Di Marco, C. Pompilio, A. Bertolino, A. Calabrò, F. Lonetti, and A. Sabetta, "Yet another meta-model to specify non-functional properties," in *Proceedings of the International Workshop on Quality Assurance for Service-Based Applications*. ACM, 2011, pp. 9–16.

[17] P. Mohagheghi and V. Dehlen, "A metamodel for specifying quality models in model-driven engineering," in *Proc. The Nordic Workshop on Model Driven Engineering*, 2008, pp. 51–65.