Nòmos: from Strategic Dependencies to Obligations

Silvia Ingolfo[1], John Mylopoulos[1], Anna Perini[2], Alberto Siena[1], and Angelo Susi[2]

[1] University of Trento, via Sommarive, 14 - Trento, Italy
{silvia.ingolfo,jm,a.siena}@disi.unitn.it
[2] Fondazione Bruno Kessler, via Sommarive, 18 - Trento, Italy
{perini,susi}@fbk.eu

Abstract. New laws are increasingly constraining informations systems.To prevent misuses of the law, requirements engineers are faced with the problem of incorporating legal prescriptions into requirements analysis. *Nòmos* is an extension of *i**, which allows to build models of legal prescriptions alongside intentional elements, and derive this way requirements that at the same time fulfill stakeholder needs and comply with relevant regulations.

1 Introduction

Over the past decades, information and communication technologies have steadily evolved, so that the concept of calculus or data processing machine has been replaced by that of a socio-technical system, consisting of software, human and organizational actors and business processes, running on an open network of hardware nodes and fulfilling vital functions for large organizations. Such systems gained the attention of governmental bodies, which are responsible for regulating them through laws, regulations and policies to ensure that they comply with security, privacy, governance and other concerns of importance to citizens and governments alike. The impact of this situation has been immense on Software Engineering as much as on business practices. It has been estimated that in the Healthcare domain, organizations have spent $17.6 billion over a number of years to align their systems and procedures with a single law, the Health Insurance Portability and Accountability Act (HIPAA), introduced in 1996[3]. In the Business domain, it was estimated that organizations spent $5.8 billion in one year alone (2005) to ensure compliance of their reporting and risk management procedures with the Sarbanes-Oxley Act[4]. In this setting, requirements engineers are faced with new challenges in eliciting requirements that at the same time fulfill the needs of stakeholders and are compliant with relevant legal prescriptions. However, unlike stakeholder requirements, which can be validated thanks to the intervention of the stakeholders themselves, requirements introduced for compliance purposes need to objectively evaluated for alignment against their originating prescriptions.

[3] Medical privacy - National standards to protect the privacy of personal health information. Office for Civil Rights, US Department of Health and Human Services, 2000.
[4] Online news published in dmreview.com, november 15, 2004.

2 Objectives

The objective of the present work is to support requirements engineers when facing domains, in which laws play a role in defining the requirements for the system-to-be. Actors are subject to legal prescriptions, which they have to adhere to, or, they may decide not to comply. To make a decision – whether to comply or not, and what tasks to undertake – it is necessary to represent both, the applicable prescriptions and the evidence of compliance, if any. Afterwards, a systematic modeling process is needed for going from an initial model of law to a set of domain-specific requirements.

The underlying issue is that the design of requirements, induced by the need to adhere to laws, and requirements, generated by rational agents, is essentially different. Rational agents do what they can to fulfill their goals. With obligations, we are assuming possibly non-cooperating agents who will not necessarily do what they can to fulfill obligations. Our objective is therefore to model the different kind of obligations for and between agents established by the laws and explore designs that include safeguards and incentives that motivate agents to fulfill their obligations.

3 Contribution

Nòmos [4, 1, 6] is a goal-oriented, law-driven framework intended to generate requirements through which a given information system can comply to a given law. Such requirements are referred to as compliance requirements. *Nòmos* is based on the *i** framework, and exploits its capability to model: the actors of a given domain; their goals and the operationalization of goals into tasks; and the strategic relations among them. In addition, *Nòmos* provides the capability to model law prescriptions and the link between intentional elements and legal elements.

Concepts. The core elements of legal prescriptions are *normative propositions* (NPs), which are the most atomic propositions able to carry a normative meaning. NPs contain information concerning: the subject, who is addressed by the NP itself; the legal modality (i.e., whether it is a duty, a privilege and so on); and the description of the object of such modality (i.e., what is actually the duty or privilege). The legal modality is one of the 8 elementary rights, classified by Hohfeld as privilege, claim, power, immunity, no-claim, duty, liability, and disability. **Claim** is the entitlement for a person to have something done from another person, who has therefore a **Duty** of doing it. **Privilege** is the entitlement for a person to discretionally perform an action, regardless of the will of others who may not claim him to perform that action, and have therefore a **No-claim**. **Power** is the (legal) capability to produce changes in the legal system towards another subject, who has the corresponding **Liability**. **Immunity** is the right of being kept untouched from other performing an action, who has therefore a **Disability**. Complex legal prescriptions are created in law documents by structuring NPs through conditions, exceptions, and other conditional elements. Such elements are captured in *Nòmos* by introducing priorities between NPs. For example, a data processor may be allowed (i.e., it has a privilege) to process the data of a subject; but the right of the subject to keep his/her data closed w.r.t. third parties has a higher priority on the privilege, thus constraining the way data is used by the processor.

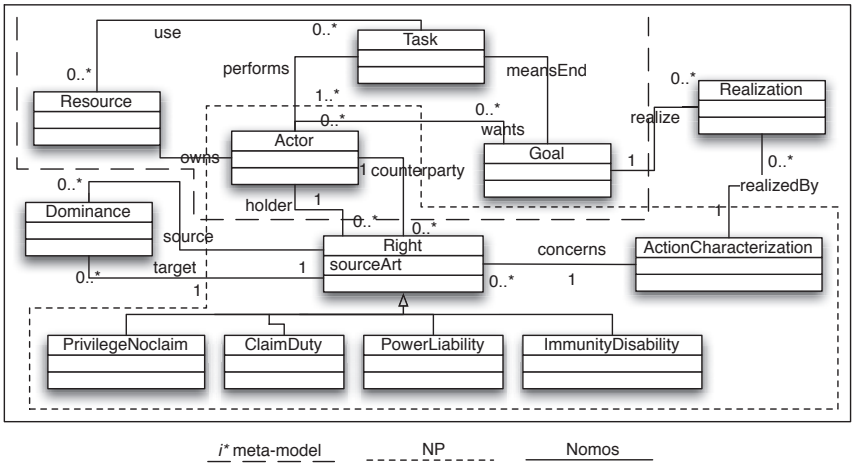

Figure 1. The meta-model of the *Nòmos* language.

Metamodel. Figure 1 depicts the *Nòmos* meta-model, and shows how it integrates the representation of NPs with the representation of goals. The dashed line contains a part of the *i** meta-model, including to the `Actor` class and its `wants` association with `Goal`. The dotted line contains the elements that form a NP. The join point between the goal-oriented and the legal part of the meta-model is the class `Actor`, which is at the same time stakeholder in the domain and subject of NPs. Actors are associated to the class `Right` by means of the `holder` relation. So on the one hand actors want goals, perform tasks, own resources; on the other hand, they are addressed by rights carried by NPs. Rights also impact on the social interaction of actors - in the Hohfeldian legal taxonomy, rights are related by correlativity relations: for example, if someone has the claim to access some data, then somebody else will have the duty of providing that data. This means that duty and claim are correlatives. Similarly, privilege-noclaim, power-liability, immunity-disability are correlatives: they *describe the same reality from two different points of view*. So instead of defining two separate classes for "duty" or "claim", we have a single class, `ClaimDuty`, which is able to model both. Similarly the classes `PrivilegeNoclaim`, `PowerLiability` and `ImmunityDisability`, each of them sub-class of the abstract class `Right`. Priorities between rights are captured in the meta-model by means of the Dominance class, which connects two rights. The `ActionCharacterization` class contains the actual object of the NP. Such prescribed action is bound to the behaviour of actors by means of the `Realization` class. It specifies that a certain goal is wanted by the actor in order to accomplish the action prescribed by law. For more information on the *Nòmos* meta-model, see [5].

Visual notation. Figure 2 exemplifies the *Nòmos* visual notation, as applied in a study about legal compliance of requirements for a healthcare information system [3]. When a patient ([User]) accesses a health care center, at the check-in the EHR of the patient has to be retrieved from the system. In the health care centre accessed by the

patient, the system (a [Local Authority]) executes a query on the local database, and the [S1] service furnishes such data. If the data is not found in the local database, the [Local Authority] forwards the request to the [S2] service, which returns the name of the reference [Certificate Authority]. The Authority is queried to have certified data. But [Certificate Authority] can also be unable to provide the requested data. In this case, the local authority contacts another Local Authority (the actor [Peer Local Authority] in the diagram), which in turn executes a local search or queries its own reference Certificate Authority. If the searched data don't exist in the system, the Local Authority proceeds inserting it, and marking it as "dirty". In this case, after the data insertion, the Local Authority invokes the [S3] service, which broadcasts the data to the whole system. When the broadcast notification is received, each Local Authority updates its local database. However, the privacy law lays down many prescriptions concerning the processing of personal data (in particular, sensitive data) of patients. For example, the law requires the owner's confirmation for the data being processed. In Figure 2, this is depicted by means of the normative proposition [Confirmation as to whether or not personal data concerning him exist]), extracted from article 7.1. The normative proposition is modeled as a claim of the patient, held towards the Local Authority, which has therefore a corresponding duty. This results in two additions to the diagram. The first one concerns the insertion of the data into the local database, and subsequent broadcast to the system. In this case, before the broadcast is executed, it is necessary to obtain the patient's authorization (goal [Ask user authorization]), and to add such information in the broadcast message. The second case concerns the reception of the broadcast system by a Local Authority. In this case, before updating the local data with the received one, the Local Authority must verify that in the broadcast message the authorization to data processing is declared (task [Verify user authorization]). This approach allows for distinguishing goals with respect to their role in achieving compliance: strategic goals are those goals that come from stakeholders and represent needs of the stakeholders; compliance goals are those goals that have been developed to cope with legal prescriptions. In the figure, the goal [Update data locally] is a strategic goal, because it is only due to the reason-to-be of the owning actor; viceversa, [Ask user authorization] is a compliance goal, because it is due to the need of complying with the [Confirmation as to whether or not personal data concerning him exist] claim of the user. So, we can infer that while the first can be dropped according to stakeholders needs, the latter can not, unless its impact on the compliance condition is evaluated.

Process. The definition of compliance we have provided before, clearly outlines that reaching compliance is an iterative process that revises the initial requirements model to guarantee that these two properties are met by the final model. The procedure we propose is structured along three logical phases [2]:

1. The *analysis phase* takes as input the model of requirements, expressed as a set of goals to be achieved and tasks to be performed by stakeholders, and a set of NPs with possible irregularities highlighted. We define as irregular a situation where either an element of the model directly violates a norm, or where an element is addressed by a regulation and therefore needs to be checked for compliance.

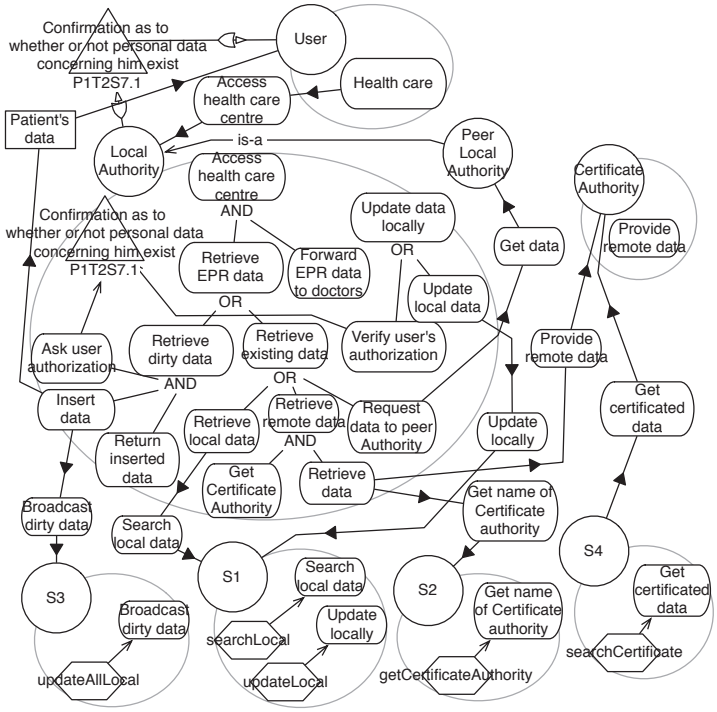

Figure 2. A goal model for the demo scenario of the Amico project.

2. The model is then followed by a *compliance check* where the criteria for compliance are evaluated. If the model is compliant, the process returns the model, else we move to the next phase.

3. The *modeling phase* aims at amending a requirements model that is not compliant. The model is expanded and revised by requirement engineers to satisfy the two compliance constraints. A discussion evaluates the acceptability and validity of the solution proposed.

Since this last step modifies the initial model, the process is iterated to ensure that no irregularities have been introduced during modeling. When a cycle is completed without introducing modifications in the solution layer – i.e., in the models – the process ends and compliance is said to be achieved. The key part of the approach is to be able to guarantee that the revisions actually make the system compliant: all the corrections made to the system — as well as the assumptions behind the corrections — are based on the fundamental concept of providing validation through argumentation.

4 Conclusions

The *Nòmos* framework has allowed the assessment of a class of problems, which were otherwise difficult to model in vanilla *i**. Specifically, it has allowed to add to the descriptions of legal prescriptions, that are not intentional elements but shape the way intentional elements are designed and analyzed.

5 Ongoing and future work

The complexity of laws and regulations dictates the need for new design and analysis techniques for software systems. The *Nòmos* meta-model currently supports a basic representation of conditional elements that are typically found in laws. It is necessary to enrich the expressiveness of this specific aspect of *Nòmos* to capture so-called *legal alternatives* — i.e., alternative ways of being compliant, which are implicit in legal prescriptions.

From a different standpoint, a key issue for the requirements compliance problem concerns the form of evidence provided that indeed a requirements model complies with a given law (fragment). Formal method techniques are generally heavy-weight in the notations they use for modeling laws and requirements, as well as in the reasoning tools they employ to establish compliance, and as a result they have not succeeded in being the proper solution to prove compliance. Alternatively, we aim at establishing compliance through argumentation among the stakeholders who state positions, e.g., "this requirement does not comply with this part of the law" and argue for or against them until (hopefully) consensus is reached.

References

1. S. Ghanavati, D. Amyot, L. Peyton, A. Siena, A. Perini, and A. Susi. Integrating business strategies with requirement models of legal compliance. *IJEB*, 8(3):260–280, 2010.
2. S. Ingolfo. Establishing compliance of software requirements through argumentation. Master's thesis, University of Trento, Italy, 2011.
3. A. Siena, G. Armellin, G. Mameli, J. Mylopoulos, A. Perini, and A. Susi. Establishing regulatory compliance for information system requirements: An experience report from the health care domain. In J. Parsons, M. Saeki, P. Shoval, C. C. Woo, and Y. Wand, editors, *ER*, volume 6412 of *Lecture Notes in Computer Science*, pages 90–103. Springer, 2010.
4. A. Siena, J. Mylopoulos, A. Perini, and A. Susi. Designing law-compliant software requirements. In *Conceptual Modeling - ER 2009*, pages 472–486, 2009.
5. A. Siena, J. Mylopoulos, A. Perini, and A. Susi. A meta-model for modeling law-compliant requirements. In *2nd International Workshop on Requirements Engineering and Law (Relaw'09)*, Atlanta, USA, September 2009.
6. A. Villafiorita, K. Weldemariam, A. Susi, and A. Siena. Modeling and analysis of laws using bpr and goal-oriented framework. In L. Berntzen, F. Bodendorf, E. Lawrence, M. Perry, and Å. Smedberg, editors, *ICDS*, pages 353–358. IEEE Computer Society, 2010.