Designing Law-Compliant Software Requirements

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

[1] FBK - Irst, via Sommarive 18 - Trento, Italy
{siena,perini,susi}@fbk.eu
[2] University of Trento, via Sommarive 14 - Trento, Italy
jm@cs.toronto.edu

Abstract. New laws, such as HIPAA and SOX, are increasingly impacting the design of software systems, as business organisations strive to comply. This paper studies the problem of generating a set of requirements for a new system which comply with a given law. Specifically, the paper proposes a systematic process for generating law-compliant requirements by using a taxonomy of legal concepts and a set of primitives to describe stakeholders and their strategic goals. Given a model of law and a model of stakeholders goals, legal alternatives are identified and explored. Strategic goals that can realise legal prescriptions are systematically analysed, and alternative ways of fulfilling a law are evaluated. The approach is demonstrated by means of a case study. This work is part of the Nomos framework, intended to support the design of law-compliant requirements models.

1 Introduction

In an ever-more complex and fluid world, there has been a steady increase in government laws and regulations, industrial standards, and company policies that need to be taken into account during the design of new organisational systems. These laws, regulations and policies need to be analysed and accommodated, somehow, during the definition of requirements for the new system. The problem of compliance to regulations is even more difficult for an existing organisation who has to restructure and reengineer its operation to achieve compliance. The problem is compounded for multi-national organisations whose systems operate in international jurisdictions where multiple, often contradictory laws apply.

The engineering/reengineering of law-compliant organisational information systems has become a major factor in IT-related projects. It has been estimated that in the Healthcare domain, organisations have spent $17.6 billion over a number of years to align their systems and procedures with a single law, the U.S. Health Insurance Portability and Accountability Act (HIPAA), introduced in 1996 [1]. In the Business domain, it was estimated that organisations would spend $5.8 billion in one year alone (2005) to ensure compliance of their reporting and risk management procedures with the Sarbanes-Oxley Act (SOX) [2].

We view the problem of compliance as a modelling problem. Laws are expressed in terms of a set of legal concepts, such as those of ”right”, ”obligation” and ”privilege”.

Requirements, on the other hand, are expressed in terms of stakeholder goals. The definition of law-compliant requirements is then a problem of transforming, through a systematic process, models of rights, obligations, privileges etc. into models of actors, goals and actor inter-dependencies. This paper proposes such a systematic process for generating law-compliant requirements, given a model of the law and a model of initial stakeholder goals. Our approach is illustrated with an example scenario of a (U.S.) hospital that needs to be compliant with HIPAA while setting up a new information system to manage service reservations. The work reported here is part of the Nomos framework presented in [16]. In earlier work, [16], we introduced a conceptual model for laws and defined the notion of compliance between a model of law and a model of system requirements. In this work, we focus on the process of generating law-compliant requirements.

The rest of the paper is structured as follows: Section 2 recalls the Nomos framework concepts and its modelling language, which is shortly illustrated on the example scenario; Section 3 describes how to build a model of law-compliant requirements starting from a model of law and a set of initial requirements; Section 4 discusses the properties of the generated requirements model; Section 5 reviews the related works; finally, Section 6 concludes.

2 Research Baseline

Nomos[3] is a modelling framework that aims at supporting requirements analysts in dealing with the problem of requirements compliance. It offers a conceptual solution that combines elements of goal orientation with elements of legal theory to argument about compliance of a certain requirements set and to *derive* models of compliant requirements, starting from a model of law.

For its nature, formal proof of run-time compliance can't be given at requirements time: there are properties of law that makes that the compliance condition can only be stated *ex-post* by the judge - e.g., the subsequent design could be wrong, people could behave differently from what is assigned to them according to their roles, software programs could be bugged and also behave differently from what expected, and finally law can be intentionally ambiguous, as pointed out in [3]. For this reason, we have introduced the concept of *Intentional Compliance* [15] as the assignment of actors responsibilities such that if every actor fulfils its goals, then law is respected. We derive a general rule to define the notion of requirements compliance. Given a set of requirements represented as actors goals, R, and a set of domain assumptions D, we say that the requirements are compliant with a law L, and write $R, D \models L$, if, for every possible state of the world, if R holds, then L holds.

Intentionality. In the above formula, R represents the sets of possible alternatives, expressed in terms of stakeholders goals. The Nomos framework adopts a security-oriented extension of the *i** modelling framework [19], namely *SecureTropos* [9], to represent stakeholders and their goals. Worth mentioning that this choice is arbitrary - other frameworks could be used or adapted to be used as well, as long as they provide

[3] From Greek $N\acute{o}\mu o\varsigma$, which means "norm".

primitives for modelling actors, goals, and security relationships between actors. The *i** framework [19] models a domain along the two following perspectives: the **strategic rationale** of the actors - i.e., a description of the intentional behaviour of domain stakeholders in terms of their goals, tasks, preferences and quality aspects (represented as softgoals); and the **strategic dependencies** among actors - i.e., the system-wide strategic model based on the relationship between the depender, which is the actor who "wants" something and the dependee, that is the actor who has the *ability* to do something that contributes to the achievement of the depender's original goals. Strategic dependencies can then be secured [9] by adding information on the trust that actors have in each other. Depending on their trust, actors can delegate the execution of plans or achievement of goals, or they can delegate the permission to use resources.

Elements of Legal Theory. The Hohfeld's taxonomy [10] is a milestone of juridical literature that proposes a widely accepted classification of legal concepts. It is grounded on the notion of ***right***, which can be defined as "entitlement (not) to perform certain actions or be in certain states, or entitlement that others (not) perform certain actions or be in certain states"[4]. Rights are classified by Hohfeld in the 8 elementary concepts of *privilege*, *claim*, *power*, *immunity*, *no-claim*, *duty*, *liability*, *disability*, and organised in opposites and correlatives. **Claim** is the entitlement for a person to have something done from another person, who has therefore a **Duty** of doing it; e.g., if John has the claim to exclusively use of his land, others have a corresponding duty of non-interference. **Privilege** (or liberty) 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**; e.g., giving a tip at the restaurant is a liberty, and the waiter can't claim it. **Power** is the (legal) capability to produce changes in the legal system towards another subject, who has the corresponding **Liability**; examples of legal powers include the power to contract and the power to marry. **Immunity** is the right of being kept untouched from other performing an action, who has therefore a **Disability**; e.g., one may be immune from prosecution as a result of signing a contract. Two rights are **correlatives** [10] if the right of a person implies that there exists another person (it's counter-party), who has the correlative right. For example, if someone has the claim to access some data, then somebody else will have the duty of providing that data, so duty and claim are correlatives; similarly, privilege-noclaim, power-liability, immunity-disability are correlatives. The concept of *correlativeness* implies that rights have a **relational nature**. In fact, they involve two subjects: the owner of the right and the one, against whom the right is held - the *counter-party*. Vice versa, the concept of *opposition* means that the existence of a right excludes its opposite.

The Nomos modelling language. The Nomos modelling language, whose meta-model is depicted in Fig. 1, conceives law as a partially ordered set of *Normative Propositions* (NP). Basically, NPs are the most atomic element in which a legal prescription can be subdivided. The core element of a NP is the hohfeldian concept of right (class `Right`). Since rights have a dual nature, the relation of "correlative" or "equivalent" means that the two rights that it connects describe the same reality, but from two different points of view. This results in 4 classes of rights, namely `PrivilegeNoclaim`, `ClaimDuty`, `PowerLiability` and `ImmunityDisability`, which subsume the 8 hohfeldian

[4] From `http://plato.stanford.edu/entries/rights/`

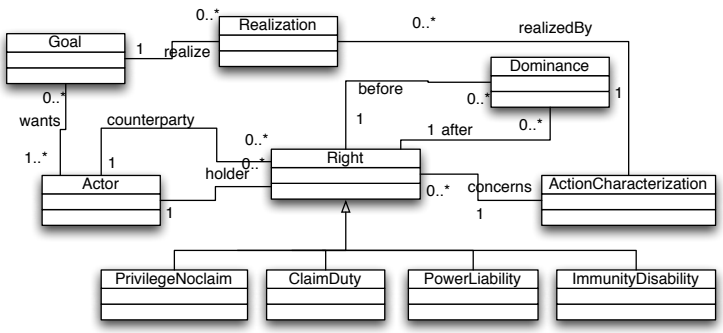

Fig. 1. The Nomos modelling language and its meta-model.

concepts. The object of rights are *actions*, (as defined in [13]), which consist in the description of either something to be done (*behavioural action*) or something to be achieved (*productive action*). In the meta-model we refer to it as `ActionCharacterization`. Finally, rights address two domain actors (class `Actor`): the right's `holder`, and its `counter-party`. For conditional elements such as exceptions, time conditions and so on we give a uniform representation by establishing an order between normative propositions. Given a set of normative propositions $\{NP_1...NP_n\}$, $NP_k > NP_{k+1}$ - read: NP_k overcomes NP_{k+1} - means that if NP_k is satisfied, then the fulfilment of NP_{k+1} is not relevant. This is captured in the meta-model via the definition of the concept of the class `Dominance`, connected to the class `Right`.

As said, the Nomos meta-model combines elements of legal theory with elements of goal orientation. In Fig. 1, a part of the *i** meta-model (taken from [17]) is also depicted. The `Actor` class is at the same time part of NPs (rights concern domain actors) and of the *i** meta-model (an actor wants goals). This way, Nomos models are able to inform whether a goal fits the characterisation given by law. In Fig. 1, this is expressed with the concept of **realisation** (class `Realization`), which puts in relation something that belongs to the law with something that belongs to the intentions of actors.

Normative propositions are represented in the Nomos frameworks by means of a visual notation, depicted in Fig. 2, that has been defined as an extension of the *i** visual notation. The actors linked by a right (holder and counter-party) are modelled as circles (i.e., *i** actors). The specified action is represented as a triangle and linked with both the actors. The kind of right (privilege/noclaim, claim/duty, power/liability, immunity/disability) is distinguished via labels on both the edges of the right relationships. Optionally, it's also possible to annotate with the same labels on the left side the triangle representing the action. The language also introduces a *dominance* relationship between specified actions, represented as a link between two prescribed actions and labelled with a "$>$" symbol that goes from the dominant action to the dominated one. Finally, a *realisation* relation is used in the language to establish a relation between one element of the intentional model and one element of the legal model.

Running Example. Title 2 of HIPAA addresses the privacy and security of health data. Article §164.502 of HIPAA says that: *(a) A CE may not use or disclose PHI,*

Src §164.	Id	Right	Holder	Counterparty	Action characterisation	Dominances
§502a	NP1	CD	Patient	CE	not DisclosePHI	-
§502a1i	NP2	PN	CE	Patient	DisclosePHI	NP1
§502a2i	NP3	CD	Patient	CE	DisclosePHI	NP1,NP2
§502a2ii	NP4	PL	Secretary	CE	DisclosePHI	NP1
§314a1ii	NP5	CD	CE	BA	no KnownViolations	NP6,NP7,NP8
§314a1ii	NP6	ID	CE	Authority	EndViolation	NP7,NP8
§314a1iiA	NP7	ID	CE	Authority	TerminateContract	NP8
§314a1iiB	NP8	ID	CE	Secretary	ReportTheProblem	-
§314a2iiC	NP9	CD	CE	BA	ReportSecurityLacks	-

Legenda: CD = Claim/Duty; PN = Privilege/Noclaim; PL = Power/Liability; ID = Immunity/Disability

Table 1. Some Normative Propositions identified in §164.314 and §164.502.

except as permitted or required by this subpart [...] (1) A covered entity is permitted to use or disclose PHI [...] (i) To the individual; (2) A CE is required to disclose PHI: (i) To an individual, when requested [...]; and (ii) When required by the Secretary. Out of this law fragment, it is possible to identify the normative propositions that compose the law fragment. The identified normative propositions are summarised in Table 1. The first row of the table contains a reference to the source text (more information can be stored here, but it is not shown in the table due to lack of space). "Id" is a unique identifier of the NP. Holder and counterparty are the involved actors. "Action characterisation" is the description of the action specified in the NP. To identify the NPs, prescribing words have been mapped in the right specifiers; e.g., "is permitted" has been mapped into a *privilege*, "is required" has been mapped into a *duty*, and so on. The name of the subjects are extracted by either using an explicit mention made by the law (e.g., "a CE is not in compliance if..."); or, when no subject has been clearly detected, by identifying who carries the interest that the law is furthering. Finally, the priority column establishes the dominance relationships between NPs. For example, an exception like the one in the first sentence ("A CE may not [...] except [...]") has been mapped into a dominance of every other proposition of §164.502 over NP1. Fig. 2 depicts a diagram of §164.314 and §164.502. The diagram is a graphical representation of the NPs listed in Table 1.

3 A Process for Generating Law-Compliant Requirements

Reasoning about goals allows to produce requirements that match the needs of the stakeholders [18, 20]. However, goals are the expression of the actors intentionality, so their alignment with legal prescriptions has to be argued. The meta-model of Fig. 1 provides a bridge between intentional concepts, such as goal, and legal concept, such as right. Here we show how to generate law-compliant requirements by means of conceptual modelling. Specifically, we assume to have an initial model of the stakeholders goals and a model of the law.

For example, we depict a scenario in which a US hospital has its own internal reservation system, consisting in the employee personnel answering phone calls and scheduling doctors appointments on an agenda. The hospital wants now to set up a new

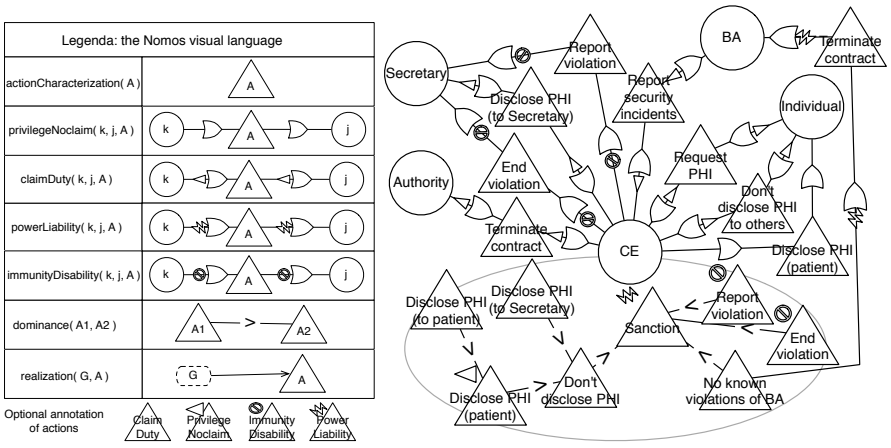

Fig. 2. The Nomos modelling languages: visual representation of §164.314 and §164.502.

information system - to manage the reservations, quickly retrieve the availability of rooms and devices in the hospitals, and ultimately optimise the reservation according to the needs of the patients and doctors - and to reduce expenses the hospital wants to outsource the call center activity to a specialised company. Since the reservation system is intended to deal also with the patients PHI, system requirements have to be carefully analysed to be made compliant with the HIPAA law described in previous section.

In this context, to generate law-compliant requirements the analyst has to answer to four types of questions:

- Which are the actors addressed by laws? And by which laws? Reconciling the stakeholders identified in the domain with the subjects addressed by law is necessary to acquire knowledge on what normative propositions actually address stakeholders.

- What does the law actually prescribes? Are there alternative possibilities to comply with a given prescription?

- How is it possible to allow actors to achieve their own goals while ensuring compliance with the law?

- How is it possible to maintain the compliance condition through the responsibility delegations that generally occur in an organisational structure?

We answer to these questions in a series of steps that form a modelling process. Starting from an initial requirements model (R) and a model of law (L) (and the proper domain assumptions, D), the process allows to generate a new requirements set, such that $R, D \models L$. The output of the process for our running example is depicted in Fig. 3. In the following, we will detail the modelling process that produces that output, describing the *why* and *how* of each step of the process, and its results.

Step 1. Bind domain stakeholders with subjects addressed by law

Why. In the Nomos meta-model of Fig. 1, actors represent the binding element between laws and goals, but during modelling this binding can't be automatically deduced. Actors wanting goals are extracted from the domain analysis, while actors addressed by

laws are extracted from legal documents. The different sources of information, as well as the different scope and interests covered, raises the need to know who is actually addressed by which law.

How. The binding is operated by the analyst, possibly comparing how actors are named in the law, with respect to how they are named in the domain analysis - or, if law identifies the addressee by recalling the most notable (intentional) elements of its behaviour, then those elements are compared with the elements of the stakeholders actors behaviour. When a domain actor is recognised to be a law subject, the corresponding rights are assigned to the actor. Actors that are not part of the domain, but that interact with other domain actors have to be added to the requirements model. Otherwise, law subjects can be excluded from the requirements model.

Result. The result of this step is a model of rights as in Fig. 2, in which actual domain stakeholders replace law subjects.

Example. The Hospital under analysis in our domain is an entity covered by the law (CE). The Patient is the actor referred to as the Individual in the law. And the Call Center in this scenario is a business associate (BA) of the covered entity. Some actors, such as the Secretary and what has been called the Authority were not introduced in the domain characterisation, but have legal relations with other actors. Finally, some actors, such as the Doctor and the Data Monitor are not mentioned in the legal documents taken into consideration.

Step 2. Identify legal alternatives

Why. Dominance relations establish a partial order between NPs such that not every NP has actually to be fulfilled. For example, a law $L = \{NP_a, NP_b, NP_c\}$, with $NP_b > NP_a$. This means that NP_b *dominates* NP_a: as long as NP_b holds, NP_a does not, and it is quite common in law. Let suppose that NP_a says that it is mandatory to pay taxes, and NP_b says that it is possible to use the same amount of money, due for taxes, to make investments. $NP_b > NP_a$ means that, if a company makes an investment, then it does not have to pay taxes for the same amount. Now, with the given NPs and dominance relations, companies have two alternatives: $L_1 = \{NP_a, NP_c\}$, and $L_2 = \{NP_b, NP_c\}$. We call these alternative prescriptions *legal alternatives*. As long as many alternative prescriptions exist, the need arises for selecting the most appropriate one. Legal alternatives can be different for a large number of NPs, which can change, appear or disappear in a given legal alternative, together with their dominance relationships, so that the overall topology of the prescription also changes. This causes the risk that the space of alternatives grows too much to be tractable, so the ultimate problem is how to cut it.

How. To solve this problem, we introduce a decision making function that determines pre-emptively whether a certain legal alternative is acceptable in terms of domain assumptions, or if it has to be discarded. The decision making function is applied by the analyst whenever a legal alternative is detected, to accept or discard it. We define four basic decision making function (but hybrid or custom functions can be defined as well):

a) Precaution-oriented decision maker. It wants to avoid every sanction, and therefore tries to realise every duty. Immunities are also realised to avoid sanctions to occur.

b) Opportunistic decision maker. Every alternative is acceptable - including those that involve law violation - if it is convenient in a cost-benefit analysis with respect to

the decision maker's goals. In a well-known example of this function, a company has decided to distribute its web browser application, regardless of governmental fines that have been applied, because the cost of changing distribution policy has been evaluated higher than the payment of the fine.

c) Risk prone decision maker. Sanctions are avoided by realising the necessary duties, but ad-hoc assumptions are made that the realised duties are effective and no immunities are needed. This is mostly the case in small companies that do not have enough resources to achieve high levels of compliance.

d) Highly conform decision maker. This is the case in which legal prescriptions are taken into consideration also if not necessary. For example, car makers may want to adhere to pollution-emission laws that will only be mandatory years in the future.

Result. The result of this step is a set of NPs, subset of L, together with their dominance relationships, which represent a model of the legal prescription that the addressed subject actually wants to comply with.

Example. Dominance relations of Table 1 define the possible legal alternatives. NP1 (Don't disclose PHI) is mandatory to avoid the sanction. NP5, No known violations, is also mandatory; however, law recognises that the CE has no control over the BA's behaviour and admits that the CE can be not able to respect this NP. To avoid being sanctioned, in case of violation the CE can perform some actions, End the violation (NP6) or Terminate the contract (NP7). So ultimately, NP6 and NP7 are alternative to NP5. In Fig. 3, the hospital adopts a risk-prone strategy. According to the law model, if a BA of the hospital is violating the law and the hospital is aware of this fact, the hospital itself becomes not compliant. It is however immune from legal prosecution if it takes some actions, such as reporting the violation to the secretary (NP Report violation). However, in the diagram the hospital does not develop any mechanism to face this possibility. Rather, it prefers to believe that the BA will never violate the law (or that the violation will never be known).

Step 3. Select the normative proposition to realise

Why. Another source of variability in law compliance consists in the *applicability conditions* that often exist in legal texts. The *applicability* of a certain NP could depend on many factors, both objective and subjective - such as time, happening of certain events, the decision of a certain actor and so on. For example, an actor may have a duty but only within a fixed period of time or only when a certain event occurs. So the problem arises, of which NP has actually to be realised.

How. Trying to exhaustively capture all the applicability conditions is hard and possibly useless for purposes of requirements elicitation. So, instead of trying to describe applicability in an absolute way (i.e., specify exactly when a NP is applicable), we describe it in *relative* terms: i.e., we describe that *if* an existing NP is actually applicable, then another NP is not applicable. More specifically, we use dominance relation between two NPs, $NP1$ and $NP2$, and write $NP1 > NP2$ to say that, whenever $NP1$ holds (is applicable), then $NP2$ does not hold.

Result. This step returns the bottom-most NP that has to be realised. I.e., if $NP1$ is still not realised, and $NP2$ is already realised, then $NP1 > NP2$ and $NP1$ is returned. If no other NP exist, it returns nothing.

Example. $NP1$ says that *"the CE may not disclose patient's PHI"*, and $NP3$ states that *"A covered entity is required to disclose patient's PHI when required by the Secretary"* - in this case, $NP1$ and $NP3$ are somehow contraddicting each other, since $NP1$ imposes the non-disclosure, while $NP3$ imposes a disclosure of the PHI. But the dominance relation between $NP3$ and $NP1$ states that, whenever both $NP3$ and $NP1$ - i.e., when the Secretary has required the disclosure, then the dominant NP prevails on the dominated one.

Step 4. Identify potential realisations of normative propositions

Why. Normative propositions specify to addressed subjects actions to be done (*behavioural actions*, according to the terminology used in [13]), or results to be achieved (*productive actions*). As they are specified in legal texts, actions recall goals (or tasks, or other intentional concepts); however, actions and goals differ as (i) goals are *wanted by* actors, whereas actions are *specified to* actors and can be in contrast with their goals; and (ii) goals are *local* to a certain actor - i.e., they exist only if the actor has the *ability* to fulfil them - while actions are *global*, referring to a whole class of actors; for example, law may address health care organisations, regardless whether they are commercial or no-profit, but when compliance is established, the actual nature of the complying actor gains importance; for the same reason, actions are an *abstract* characterisation of a whole set of potential actions as conceived by the legislator. It becomes so necessary to switch form the point of view of the legislator to the point to view of the actor.

How. Given a normative proposition NP that specifies an action A_{NP}, a goal G is searched for the addressed actor, such that: (i) it is *acceptable* by the actor, with respect to its other goals and preferences; (ii) the actor is known to have, or expected to have, the ability to fulfil the goal; and (iii) there is at least one behaviour that the actor can perform to achieve the goal, which makes NP fulfilled. In the ideal case, every behaviour that achieves G also fulfils NP; we write in this case $G \subseteq NP$. Otherwise, G is decomposed to further restrict the range of behaviours, until the above condition is ensured. If it is not possible to exclude that $G \nsubseteq NP$, then G is considered *risky* and the next step (Identify legal risks) is performed.

Result. If found, G (also if it is risky) is put in realisation relation with NP and becomes the top *compliance goal* for NP.

Example. One of the assumptions made for building the diagram of Fig. 3 is that the requirements analysis concerns only the treatment of electronic data. As such, from the point of view of the hospital the non-disclosure duty (NP Don't disclose PHI) is fulfilled if the PHI is not disclosed *electronically*. In the diagram, for the hospital a well-designed set of policies for accessing electronic data (goal policy-based data access) is enough to have the duty realised. This may be true, or may be too simple-minded, or may need further refinement of the goal. This is part of the modelling activity.

Step 5. Identify legal risks

Why. At organisational level, risks have a negative impact on the capability of the organisation to achieve its goals. Using *i** , risks can be treated with risk management techniques that allow to minimise them [4]. For organisations, law is also a source of a particular type of risk, or *legal risk*, which "includes, but is not limited to, exposure to fines, penalties, or punitive damages resulting from supervisory actions, as well as

private settlements"[5]. Legal risk comes from the fact that compliance decisions may be wrong, incomplete or inaccurate. In our framework, the "realisation" relation that establishes the link between a NP and a goal can't prevent legal risks to arise: for example, a wrong interpretation of a law fragment may lead to a bad definition of the compliance goal. Legal risk can't be completely eliminated. However, the corresponding risk can be made explicit for further treatment.

How. Specifically, when a goal is defined as the realisation of a certain NP, a search is made in the abilities of the actor, with the purpose of finding other intentional elements of its behaviour that can generate a risk. Given a certain risk threshold ϵ, if the subjective evaluation of the generated risk is greater than ϵ, then the risky element has to be modelled.

Result. If some of the requirements may interfere with the compliance goals, then the requirements set is changed accordingly and the new set is returned. If no risky goals have been identified, the requirements set is not changed.

Example. In Fig. 3, we have depicted the need for the hospital to have a hard copy of certain data: it's the goal Print data (assigned to the hospital for sake of compactness). If doctors achieve this goal to print patients PHI, this may prevent the use of a policy-based data access to succeed in the non-disclosure of PHI. This is represented as a negative contribution between Print data and Policy-based data access. To solve this problem, a new goal is added: Prevent PHI data printing, which can limit the danger of data printing. (Notice that here we don't further investigate how PHI printing prevention can actually be achieved.)

Step 6. Identify proof artefacts

Why. During the requirements analysis we aim at providing evidence of *intentional compliance*, which is the assignment of responsibilities to actor such that, if the actor fulfil their goal, then compliance is achieved. *Actual* compliance will be achieved only by the running system. However, in a stronger meaning, compliance can be established only *ex-post* by the judge, and at run-time this will be possible only by providing those documents that will prove the compliance.

How. After a compliance goal is identified, it can be refined into sub-goals. The criterion for deciding the decomposition consists in the capability to identify a proof resource. If a resource can be identified, then such a resource is added to the model; otherwise, the goal is decomposed. The refinement process ends when a proof resource can be identified for every leaf goal of the decomposition tree.

Result. The result of this step is a set of resources that, at run-time, will be able to prove the achievement of certain goals or the execution of certain tasks.

Example. In Fig. 3, the NP Don't disclose PHI is realised by the goal Policy-based data access, which can be proved to keep the PHI not disclosed by means of two resources: the Users DB and the Transactions report.

Step 7. Constrain delegation of goals to other actors

Why. To achieve goals that are otherwise not in their capabilities, or to achieve them in a better way, actors typically delegate to each other goals and tasks. When an actor delegates a strategic goal, a weakness arises, which consists in the possibility that the

[5] Basel Committee on Banking Supervision 2006, footnote 97

delegatee does not fulfil the delegated goal. If the delegated goal is intended to realise a legal prescription, this weakness becomes critical, because it can generate a non-compliance situation. As such, law is often the source of the security requisites that a certain requirements model has to meet.

How. Specifically, three cases exist for delegation:

1. Compliance goals. Goals that are the realisation of a NP, or belong to the decomposition tree of another goal that in turn is the realisation of a NP, can be delegated to other actors only under specific authorisation.
2. Proof resources. We have highlighted how the identification of proof resources is important for compliance purposes. The usage of proof resources by other actors must then be permitted by the resource owner.
3. Strategic-only goals. Goals that have no impact on the realisation of NPs, can be safely delegated to other actors without need to authorise it.

Result. The result of this activity is a network of delegations and permissions that maintain the legal prescriptions across the dependencies chains.

Example. In Fig. 3, the hospital delegates to the doctors the PHI disclosure to the patients. However, the hospital is the subject responsible towards the patient to disclose its PHI. This means that a vulnerability exists, because if the doctor does not fulfil its goal then the hospital is not compliant. For this reason, using the security-enhanced i^* primitives offered by SecureTropos, in the model we have to reinforce the delegation by specifying the trust conditions between the hospital and the doctor (refer to [9] for a deeper analysis on trust, delegation and permission).

4 Results and discussion

The described process results in a new requirements set, R', represented in Fig. 3 as an extended i^* model (i.e., the i^* primitives are interleaved with the Nomos and SecureTropos ones), which presents some properties described in the following.

Intentional compliance. The realisation relations show the goals that the actors have developed to be compliant with the law. As said in Section 2, these goals express the *intentional compliance* of the actor, which ultimately refers to the *choices* that are made during the requirements analysis phase. In our example, the hospital under analysis has developed 3 goals due to the legal prescriptions: Delegate doctors to disclose PHI to patients, Policy-based data access and Electronic clinical chart. Notice that the last one is optional and the hospital may choose a different alternative. Notice also that the compliance through the mentioned goals is a *belief* of the hospital, and we don't aim at providing formal evidence of the semantic correctness of this belief.

Strategic consistence. For arguing about compliance, we moved form an initial set of requirements, R. The compliance modelling algorithm basically performs a reconciliation of these requirements with legal prescriptions. The process steps described above implicitly state that, in case of conflicts between NPs and actors goals, compliance with NPs should prevail. However, if a compliance alternative is strategically not acceptable it is discarded. Therefore, if R' is found, then it is consistent with the initial requirements R.

Documentable compliance. If L' is a legal alternative for the law L chosen applying the decision making function, for all NP (addressing actor j) and for every leaf goal, there exists a set of resources, called proof resources, with cardinality ≥ 1. In the example, the intentional compliance achieved by the hospital is partially documentable through the resources Access log, Users DB and Transactions report. However, the prevention of data printing can't be documented according to the goal model, which should therefore be further refined.

Traceability. Speaking of law compliance it is important to maintain traceability between law's source and the choice made to be compliant. In case of a change in the law, in the requirements, or just for documentation purposes, it is necessary to preserve the information of where does a certain requirement come from. Having an *explicit* model of law, and having an *explicit* representation of the link between goals and NPs (the "realisation" relationship), full traceability is preserved when modelling requirements, also through refinement trees and delegation chains. For example, the delegation to the data monitor to Monitor data usage can be traced back to the decision of the hospital to Monitor electronic transactions, which in turn comes from the decision to maintain a Policy-based data access, which is the answer of the hospital to the law prescribing to keep patients PHI not disclosed.

Delegations trustworthiness. Delegations of compliance goals to other actors are secured by means of trust information plus the actual delegation to achieve goals. If this information is missing, then a security hole exists. In our example, the decision to delegate to the data monitor to Monitor data usage depends on a compliance decision (the goal Policy-based data access); if the data monitor fails in achieving its goal, then the compliance of the hospital can be compromised. So, delegating the monitoring to it causes a weakness in the compliance intentions of the hospital.

Legal risk safety. Having made explicit every goal that is intended to achieve compliance The requirements set R' contains a treatment for legal risks that arise from compliance decisions. In Fig. 3, the delegation to doctors to Disclose PHI to patients needs to be secured, since doctors are not addressed by a specific responsibility prevent the PHI disclosure, as the hospital is. Notice that delegations' trustworthiness is not addressed by our framework, and we rely on other approaches for this.

Altogether, these properties as well as the capability to *argue* about them, represents a prominent **advantage** of the framework. However, worth mentioning that our approach is not without **limitations**. Not every kind of normative prescriptions can be successfully elaborated with the Nomos framework. The more norms are technically detailed - such as standards or policies - the less our framework is useful, since technical regulations leave small margin to alternatives and discretion. Furthermore, it's important to stress the fact that the modelling framework and the process we propose is not fully automated; it needs the intervention of the analyst to perform some steps, under the assumption that performing those steps results a support for the analyst itself. More experience with its usage may possibly be converted in further refinement of the approach. Finally, complex aspects of legal sentences, such as time or exceptions, are not addressed by our framework, which ultimately focuses on alternatives exploration and selection through goals - notice that this lack could be a limitation, or an advantage, depending on the needs of the analyst.

5 Related works

Anton and Breaux have developed a systematic process, called semantic parameterisation, which consists of identifying in legal text restricted natural language statements (RNLSs) and then expressing them as semantic models of rights and obligations [5] (along with auxiliary concepts such as actors and constraints). In [12], a somehow similar approach is presented, which however takes into consideration the separation between law and requirements sentences, with the purpose of comparing their semantics to check for compliance.

Secure Tropos [8] is a framework for security-related goal-oriented requirements modelling that, in order to ensure access control, uses strategic dependencies refined with concepts such as trust, delegation and permission, to fulfil a goal, execute a task or access a resource, as well as ownership of goals or other intentional elements. We use that framework to ensure that compliance decisions, once made, are not compromised through the delegation chains in an organisational setting. The main point of departure of our work is that we use a richer ontology for modelling legal concepts, adopted from the literature on law. Models based on the law ontology allow to reason about where and how do compliance properties of requirements are generated. Along similar lines, Darimont and Lemoine have used KAOS as a modelling language for representing objectives extracted from regulation texts [6]. Such an approach is based on the analogy between regulation documents and requirements documents. Ghanavati et al. [7] use GRL to model goals and actions prescribed by laws. This work is founded on the premise that the same modelling framework can be used for both regulations and requirements. Likewise, Rifaut and Dubois use *i** to produce a goal model of the Basel II regulation [11]. Worth mentioning that the authors have also experimented this goal-only approach in the Normative *i** framework [14]. That experience focussed on the emergence of implicit knowledge, but the ability to argue about compliance was completely missing, as well as the ability to explore alternative ways to be compliant.

6 Conclusion

In this paper we addressed the problem of generating a set of law-compliant requirements for a new system, starting from a model of the laws under consideration and a model of stakeholders' original goals. A systematic process has been defined, which consists of specific analysis steps that may be performed iteratively. Each step has been illustrated along a running example. Moreover, relevant properties of the resulting requirements model have been discussed. This research is part of the Nomos framework, whose conceptualisation has been previously introduced in [16]. Further work is ongoing including a formalisation of the compliance condition and evaluation of the Nomos framework on larger case studies.

References

1. Medical privacy - national standards to protect the privacy of personal health information. Office for Civil Rights, US Department of Health and Human Services, 2000.

2. Online news published in dmreview.com, november 15, 2004.
3. Annie I. Anton and P.N. Otto. Addressing legal requirements in requirements engineering. In *IEEE Requirements Engineering Conference (RE'07)*, 2007.
4. Yudistira Asnar and Paolo Giorgini. Modelling risk and identifying countermeasure in organizations. pages 55–66, Samos, Greece, August 2006. LNCS 4347.
5. Travis D. Breaux, Matthew W. Vail, and Annie I. Anton. Towards regulatory compliance: Extracting rights and obligations to align requirements with regulations. In *14th IEEE Requirements Engineering Conference (RE'06)*, pages 49–58, Washington, DC, USA, September 2006. IEEE Society Press.
6. Robert Darimont and Michel Lemoine. Goal-oriented analysis of regulations. In Régine Laleau and Michel Lemoine, editors, *ReMo2V, held at CAiSE'06*, volume 241 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2006.
7. Sepideh Ghanavati, Daniel Amyot, and Liam Peyton. Towards a framework for tracking legal compliance in healthcare. In John Krogstie, Andreas L. Opdahl, and Guttorm Sindre, editors, *CAiSE*, volume 4495 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2007.
8. Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Requirements engineering meets trust management: Model, methodology, and reasoning. In *ITRUST-04*, volume 2995 of *LNCS*, pages 176–190. SVG, 2004.
9. Paolo Giorgini, Fabio Massacci, John Mylopoulos, and Nicola Zannone. Modeling security requirements through ownership, permission and delegation. In *IEEE Requirements Engineering Conference (RE'05)*, pages 167–176. IEEE Computer Society, 2005.
10. Wesley Newcomb Hohfeld. *Fundamental Legal Conceptions as Applied in Judicial Reasoning*. Yale Law Journal 23(1), 1913.
11. Andre Rifaut and Eric Dubois. Using goal-oriented requirements engineering for improving the quality of iso/iec 15504 based compliance assessment frameworks. In *RE '08: Proceedings of the 2008 16th IEEE International Requirements Engineering Conference*, pages 33–42, Washington, DC, USA, 2008. IEEE Computer Society.
12. Motoshi Saeki and Haruhiko Kaiya. Supporting the elicitation of requirements compliant with regulations. In *20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, pages 228–242, 2008.
13. Giovanni Sartor. Fundamental legal concepts: A formal and teleological characterisation. *Artificial Intelligence and Law*, 14(1-2):101–142, April 2006.
14. Alberto Siena, Neil A. M. Maiden, James Lockerbie, Kristine Karlsen, Anna Perini, and Angelo Susi. Exploring the effectiveness of normative i* modelling: Results from a case study on food chain traceability. In *20th International Conference on Advanced Information Systems Engineering (CAiSE'08)*, pages 182–196, 2008.
15. Alberto Siena, John Mylopoulos, Anna Perini, and Angelo Susi. From laws to requirements. In *1st International Workshop on Requirements Engineering and Law (Relaw'08)*, 2008.
16. Alberto Siena, John Mylopoulos, Anna Perini, and Angelo Susi. The Nomos framework: Modelling requirements compliant with laws. Technical Report TR-0209-SMSP, FBK - Irst, `http://disi.unitn.it/asiena/files/TR-0209-SMSP.pdf`, 2009.
17. Angelo Susi, Anna Perini, John Mylopoulos, and Paolo Giorgini. The Tropos metamodel and its use. *Informatica (Slovenia)*, 29(4):401–408, 2005.
18. Axel van Lamsweerde and Emmanuel Letier. Handling obstacles in goal-oriented requirements engineering. *IEEE Transactions on Software Engineering*, 26(10):978–1005, 2000.
19. Eric Siu-Kwong Yu. *Modelling strategic relationships for process reengineering*. PhD thesis, Toronto, Ont., Canada, Canada, 1996.
20. Pamela Zave and Michael Jackson. Four dark corners of requirements engineering. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 6(1):1–30, 1997.

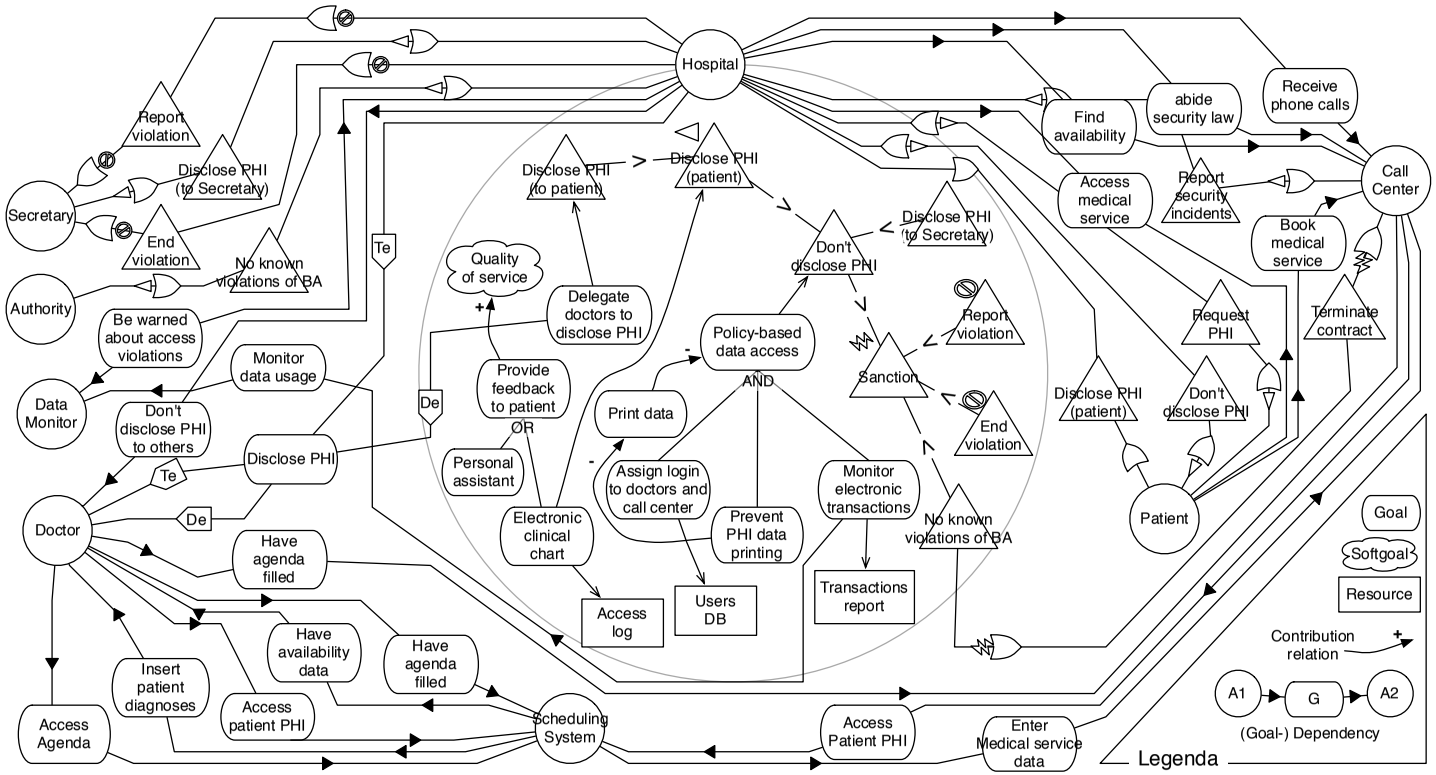

Fig. 3. A goal-oriented model of law-compliant requirements.