Enhancing Law Modeling and Analysis: using BPR-Based and Goal-Oriented Frameworks

Komminist Weldemariam, Adolfo Villafiorita, Angelo Susi
Center for Information Technology
Foundation Bruno Kessler (FBK)
Via Sommarive 18, Trento, 38123 Italy
Email: {sisai,adolfo,susi}@fbk.eu

Alberto Siena
Faculty of Computer Science
Free University of Bolzano
Bolzano, Italy.
Email: alberto.siena@unibz.it

***Abstract*—Legal documents contain regulations and principles at different levels of abstraction. They constitute rich sources of information for public administrations (PA) redesign and eventually for the software delivery that must comply with normative regulations that are specified in laws and procedures. In order to facilitate the alignment between these elements, systematic methods and tools automating regulations modeling and analysis must be developed. In this paper, we propose the integration of process modeling (named VLPM) and goal-oriented (named Nòmos) tool-supported methodologies to systematically model and analyze laws and procedures in public administration. We show that such integrated view would provide a framework that allows tracing and reasoning either top-down, from the principles to the implementation or, vice versa, bottom-up, from a change in the procedure to the principles. Finally, we also believe that this would provide a facility for interchanging models among different tools and for sharing models among different actors.**

***Keywords*—BPR; goal-orientation; laws & procedures; Nòmos; public administrations; regulation compliance; VLPM.**

I. Introduction

The introduction of new systems and procedures requires a careful modeling and analysis of laws to ensure that no conflict arises between the way things are done (i.e., processes) and the way things are meant. In that context, this paper provides our efforts in modeling and analysis of laws and procedures, which has its roots in process modeling and goal-oriented frameworks. This work is based on a conference paper on the International Conference on Technical and Legal Aspects of the e-Society 2010 (CYBERLAWS'10) [1].

Typically, there are three elements on which governments can operate to improve their public administrations (PA). One important contributor to the efficiency and improvement of PAs is the way in which the processes is (re)designed and developed. The use of Business Process Re-engineering (BPR) in that respect has become one of the recent trends undertaken to redesign processes, reduce costs and improve citizens' participation in favor of PAs [2], [3], [4]. However, there is a need to link procedures to the regulations by which procedures are defined and directed within legal documents, which contain information vary with respect to the levels of abstraction. They also describe principles or general rules that have to be followed, and thus requiring the implementation of related processes.

Another important fact to mention is the social relevance of information systems, which is determined by the way the information system is initially conceived. If misaligned with legal prescriptions, a functionality of the system can violate the rights of users, thereby breaching the law [5]. Note, however, that the system itself is not responsible for the breach, but rather, it is the owner, the designer and/or the operator who are responsible for the breach. Preventing this situation to happen is in the hand of those who are called to define the system's functionalities: the requirements analysts.

In fact, there are various broad approaches that can be used to mitigate (part of) the mentioned challenges, e.g., see in [6], [7], [8], [9], [10]. Different but complimentary approaches are often preferable for different types of challenges within a domain, and a combination of approaches would sometimes be desirable. More specifically, the integration of tool-supported methodologies with the aim of supporting the different levels of abstraction in PA processes can make easier the modeling and analysis of laws and procedures. One way to do this, for instance, is by using tools and techniques to model and analyze the underline low-level concepts of the laws as a business process. With different approach, move the emphasis of the modeling and analysis of the principles and procedures to a higher-level abstraction by interpreting each individual piece of information extracted from the law or principles as a root goal that can be decomposed into one or more subgoals. The results of these approaches would then be assessed, refined, and integrated systematically.

Along this direction, previously we presented an approach that takes the advantages of two existing tool-supported methodologies to assist the different aspects of law modeling and analysis. The first approach is based on goal-oriented framework —named Nòmos [11], [5]. The second approach, whereas uses the notion of process modeling based on subset of UML diagrams —named VLPM [12], [13]. Both approaches offer related tools to support their methodology and to allow traceability between the law and the corresponding models at different level of abstractions. We also showed that they individually have significant limitations. For example, the VLPM does not provide notations and means to represent the principles behind the procedure and to reason about possible alternative implementation; instead, Nòmos does not provide low-level implementation of the actual processes impeding the analysis of some components of the law.

This paper extends the work presented in [1] by further elaborating the context of the two approaches. We detail the underlying meta concepts and the current improvement of the two frameworks individually and as combined in handling (some of) the peculiarities in modeling and analysis of laws. For example, the VLPM includes semantic knowledge through the use of ontology. Its model sharing is now general enough since all the information can easily be stored as RDF statements [14] to maintain links between parts of the documents, parts of the process models and also elements of other types of models. It is exactly this connection that adds value to the solution we propose and makes our approach more significant than the simple juxtaposition of the two techniques. We also provide a proof of concept of the advantages we can get by putting the two together with an example.

The paper is organized as follows. The next section discusses the BRP context of modeling and analysis of PAs in connection with laws. Section III discusses the goal-oriented approach for law compliance by detailing how such processes using the Nòmos framework can be realized at higher-level. Section IVdiscusses the VLPM approach for supporting PAs with new extensions. We discuss our attempt to support BPR using the best out-of these two techniques with example in Section V. Related work, and conclusion and possible future directions are provided in Sections VI and VII.

II. Motivation

In recent years, new laws have been enacted to explicitly regulate sensitive information related to businesses and healthcare. Existing laws have been revised and also gained new meaning when referred to an Internet-based activity. As a result of this, concerns like security, privacy and governance are increasingly the focus of (digital) government regulations around the world. This trend has also created challenges in the definition and redesign of public administration (PA) processes and in the compliance of regulations. Consequently, different entities —such as PA officers, lawmakers, software engineers— are required to ensure that their software delivery complies with relevant regulations, either through (re)design or (re)engineering activity of a particular project.

In fact a decision in (any) project preliminary phase has more relevant effects than those delayed to the subsequent ones [15]. In the same way, normative choices and changes in PA influence the law effective applicability with respect to the desired system. In principle, we distinguish three different kinds of reengineering projects:

1) *System automation level.* The goal of this kind of project is introducing a new system to better support one elementary task or limited procedure. Typically small in scope, these kinds of projects provide limited improvements but are simple to implement, since they do not affect neither the procedures nor the laws.
2) *Departmental level.* The goal of this kind of project is changing the way in which work is performed within a functional unit, (often) to make it more rational and efficient. These kinds of projects are more impacting, as they require some kind of re-organization of the work, often accompanied by the introduction of new ICT systems. The impact on the laws, however, is null or minimal.
3) *Inter-departmental level.* The goal of this kind of project is providing a better implementation of those processes that involve different departments or possibly change the allocation of responsibilities or both. It is the case, for instance, of decentralization projects, where competences are moved from central government to districts. These kinds of projects are clearly the most impacting, since they act at all levels, including the normative one.

The first kind of project is a standard software development project for which there is a rich choice of tools, development cycles, and project implementation alternatives. However, the other two kinds of projects pose two peculiar and closely related challenges, which root is in the relationship between the laws and the procedures that implement such laws. These challenges are particularly common in PA domain, which are

- *Laws provide the framework that constrains and limits possible choices and alternatives in reengineering processes*. Providing tools and notations can allow to explicitly model and reason about the alternatives and constraints, and as the same time could help to develop more efficient solutions.
- *Laws and processes are intertwined as requirements and implementation are in software development processes.* Providing tools to explicitly trace the connection between laws and process elements helps for a more efficient and coherent management of the system. This can help ensuring that procedures correctly implement the law, and at the same time it could help to understand which laws might be affected by a change in the processes.

Interventions usually require to change part of the law. However, in order to understand where and how to modify the law we have to set, prepare, and validate new processes as well as to recognize the new roles and people responsibilities. Additionally, regulating a complex system or a new one requires to understand about the procedures to activate in order to answer questions like who is in charge of, when the task should be done, how to face exceptional situations, and so on. These all call a significant reform needed to provide correct snapshot about the existing procedures, to propose the (re)design and development of a new system.

As said before, one of the tools to enact this reform is the application of BPR techniques. In PA, this is an activity which involves (independently or in collaboration) law-makers who amend laws, process designers who try to optimize existing processes, and software developers, to support existing processes and/or procedures with technology. Modeling facilitates the communication and understanding of the actual organization among these users and is helpful in building a shared vision between domain experts and technicians. It also provides an easier way of analysis in order to evolve towards efficient and higher quality processes, if not pose related risks.

Unfortunately, most of the existing modeling techniques were developed with the aim of optimizing supply chains or production. This indicates that there are no as such clear goals in modeling public workflows [4]. Thus, on one hand, it is essential to know the constraints established by the law, but, on the other hand, it is also necessary to support the office in charge of such processes in order to decide the changes required by the new processes. However, one of the major difficulties encountered in this domain is the strong dependency between processes and laws. As said before, any implementation of software delivery requires a parallel action on both the redesigning of processes and on the introduction of law changes. This means that the current law (in a sense: rather than the processes), must be considered as the constraint, the engine, and the target of the reengineering activity. This link between models and laws raises further issues related to the maintainability of the models over time, since it is necessary to guarantee coherence of the models with the laws in order to have the models retain their value. We argue that these situations can reasonably be tackled by providing homogeneous and structured models of the current processes (the business architecture), which in turn should allow to redesign the new software delivery.

As we hinted in [1], it is important to devise a methodology and tool that should help tackling the two aforementioned challenges. In fact, we can apply techniques (e.g., goal-oriented methodology) that can help tackling the first problem by providing precise notations and alternatives to avoid misinterpretation and resolve ambiguities that can arise, and by performing high-level formal reasoning. The Nòmos framework [11], [5] fits for this purpose. In contrast, the second challenge can be tackled through a proper BPR approach —namely, by devising process modeling and redesigning methodology and by developing its supporting tool. Thus, the VLPM —a tool we developed for the purpose, by extending work [16]— helps tackling the second challenge.

III. Goal-oriented Approach for Law Compliance

When facing law we need to know the concepts used by law to give prescriptions. Law 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 [17]. 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 (see Table I).

TABLE I
The Hohfeldian taxonomy.

Legal relation	Opposite	Correlative
Claim	Noclaim	Duty
Privilege	Duty	Noclaim
Power	Disability	Liability
Immunity	Liability	Disability

Notice that in commonsense we might call *right* as a duty or a liability, since the word has a slightly different meaning. Here the more intuitive meaning of "right" is a **Claim**, which is the entitlement for a person to have something done from another person who has therefore a **Duty** of doing it. For example, 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**. For example, giving a tip at the restaurant is a liberty, and the waiter cannot 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**. For example, one may be immune from prosecution as a result of signing a contract.

Two rights are **correlatives** [18] if the right of a person A implies that there exists another person B (A's counterparty), 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 *counterparty*. Vice versa, the concept of *opposition* means that the existence of a right excludes its opposite.

The choice to adopt the Hohfeldian taxonomy of rights is due to several factors. First, as said above, the importance it has in the juridical literature suggests that this is actually the kind of information that we need to know about law. Second, Hohfeldian has a range of concepts and level of abstraction. These make its representation capabilities very close to the expressiveness of legal texts. In fact this consideration mainly comes from experience: constructs like powers, immunities and so on do actually exist in legal texts. Differently, for example, from deontic logic-based approaches, the proposed taxonomy is able to successfully capture them. Finally, the Hohfeldian concepts that have a *descriptive* nature, rather than *prescriptive*, acting as the bridge between the world of the "ought", typical of legal prescriptions, and the world of domain description. However, the Hohfeldian concepts do not prescribe what stakeholders should do, but rather, they describe what are the legal relations that bind them. This is of particular importance in requirements engineering, whose first step (early requirements analysis) is to describe the so-called "as-is", before specifying the "to-be". So, linking a description of stakeholders' goals with a description of applicable laws can allow to reason about compliance and compliant alternatives. In the following, we (formally) characterize such a link.

Rights are not symmetric: the position of the *claim* owner is different from the position of the *duty* owner. Generally speaking, the two positions are called active (juridical) position and passive (juridical) position, and each right has exactly an active position and a passive position. To capture this characteristic in the meta-model, active `Actors` are in `holder` relation, while passive `Actors` are in `counterparty` relation with respect to the right (see Figure 1).

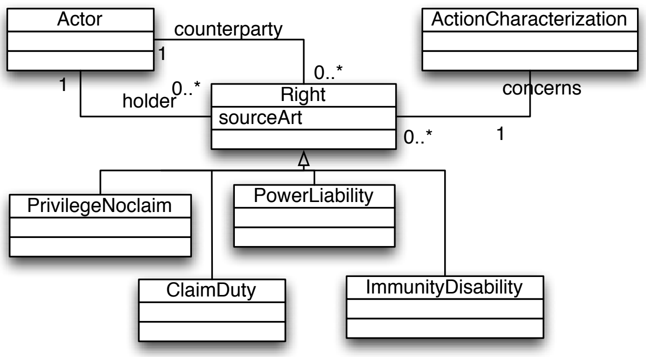

Fig. 1. The Nòmos metamodel: elements of a normative proposition.

The last component of a normative proposition is called *action*, the actual object of the right. An "action" designates a description of the set of admissible states of the world. To avoid confusion with a more common use of such word, we refer to it as `ActionCharacterization`. Each `Right` is in `concerns` relations with exactly one `ActionCharacterization`, but an `ActionCharacterization` can be addressed by a number of rights, as depicted in Figure 1.

A. Modeling the Structure of Law

The concept of normative proposition allows to split the complexity of legal statements into atomic elements. But the legal prescriptions contained in laws have more properties that have to be considered. In particular, legal prescriptions are articulated structures built with conditions, exceptions, and so on. It is important to capture the effects of these conditionals in order to obtain a meaningful requirements set. We give a uniform representation of conditional elements by establishing an order between normative propositions. For example, a citizen may have the duty to give his personal details to the policemen. However, if the policeman does not identify itself correctly as a policeman, then the citizen is free whether to do it or not. Instead of trying to formalize the *if [. . .] then [. . .]* condition, we split the problem in three steps. First, we define a first right, $r1$, —a duty of the citizen —to give personal details to the policemen. Second, we define another right, $r2$, —a privilege of the citizen —to give personal details to other citizens. Finally, we establish an order between the two: $r1 > r2$ which means that, whenever $r1$ is applicable, $r2$ is not. This is captured in the meta-model of Nòmos by the concept of **dominance** (class `Dominance`), as shown in Figure 2. This is connected to the class `Right`, which establishes the priority of the *source* right over the *target* one.

The normative propositions, manually extracted and ordered according to the meta-model, are put together to form the

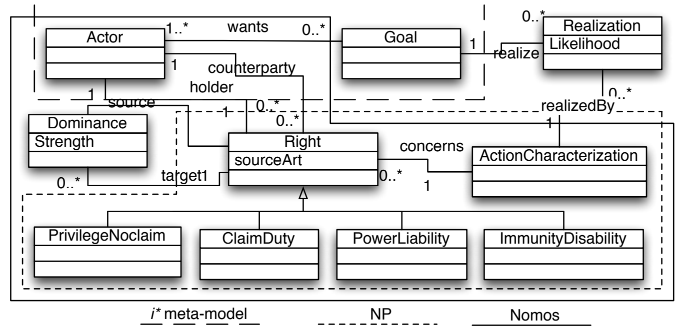

Fig. 2. The link between rights and goals as proposed by the Nòmos meta-model.

model of the law. As the meta-model shows, such a model does not contain anymore information on the physical structure of the law given by its nesting into titles, paragraphs and so on. Moreover, it it does not contain cross-reference information. However, to each right (class `Right`) carried by a normative proposition, we are able to associate its source article or any further information to precisely record where does the normative proposition come from. This would allow to ensure full traceability, as pointed out in the following.

B. Bridging Law and Requirements

Existing requirements engineering frameworks generally rest on the idea of deriving the requirements for a software system from the analysis of the stakeholders' goals that the system-to-be will support once developed and deployed. This approach has demonstrated to be effective in successfully capturing strategic requirements. However, it hardly applies to the need of arguing about the compliance of such strategic requirements with the legal ones. As already pointed out, rights concern actions which intuitively are descriptions of the behavior wanted from the addressee actor, and can result in a goal or task of that actor. However, an action characterization in itself is not a goal neither a task for two main reasons. First, a goal is a state of the world *wanted* by an actor, whereas an action characterization is a state of the world *imposed* to the actor. Second, an action is a state of the world prescribed to an *abstract* actor —a *class* of actors, and as such it is also a *class* of actions. A goal, whereas is a specific state of the world wanted by a specific actor. This makes necessity to separate the concept of goal from the one of action characterization to avoid misleading shortcuts.

In Nòmos, to describe the concepts of the strategy, we adapt the *i** modeling framework [19]. Specifically, we use the *i** version as defined for the Tropos methodology [20]. Worth mentioning that this choice is arbitrary —other frameworks could be used or adapted to be used as well, as long as they provide primitives for modeling actors, goals, and relationships between actors. The *i** framework models a domain along two perspectives: the *strategic rationale* of the actors —i.e., a description of the intentional behavior 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, in a given organizational setting, 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. An actor has the ability to achieve a goal if the actor has in the set of intentional elements that characterize it (such as sub-goals, tasks and resources) an element or a set of elements whose purpose is the achievement of the goal; or, if the actor can delegate the achievement of the goal to another actor — i.e., the dependee. The concept of ability is important because it allows to understand what are the characteristics of the specific actor existing in the domain, w.r.t the abstract actor addressed by the law.

With these ingredients, we are able to establish if a certain goal or task fits the characterization given by law, and to represent it in the model. In Figure 2, this is expressed with the concept of **realization** (class `Realization`), which puts in relation something that belongs to the law with something that belongs to the intentions of actors. This will be the starting point to argue about compliance of requirements models with law.

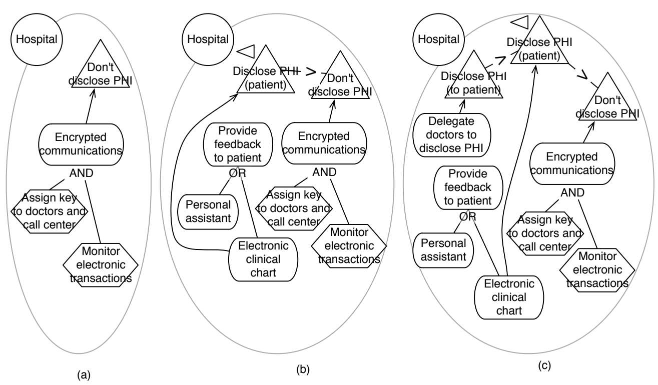

Fig. 3. An example: Law and Strategic Modeling using the Nòmos modeling language.

Figure 3(a) depicts excerpt model of a law fragment taken form the U.S. Health Insurance Portability and Accountability Act (HIPAA), and the goals’ fulfilling law and tasks that operationalize such goals. The fragment contains the duty imposed to hospitals to keep patients’ Personal Health Information (PHI) to be closed. The duty is fulfilled by the Hospital by setting up an encrypted electronic communication mechanism, which in turn is refined into the two leaf level tasks “Assign key to doctors and call center” and “Monitor electronic transaction”. This choice means that the running system will need to support its processes with these activities. Alternatively (Figure 3(b)), the law gives hospitals the possibility to disclose patients’ PHI, if the receivers are the patients themselves. The introduction of this new principle involves a possible change in the underlying processes supported by the system-to-be. Similarly, as in Figure 3(c), the introduction of the last principle —a duty, for the Hospital, to disclose such information to the patient upon request— further impacts on the supported processes, as it requires the hospital to receive the requests and to delegate somebody to disclose the data.

IV. Enhanced Law Modeling With VLPM

This section presents an approach where process models for procedures are modeled, and changes in laws are mapped in the models in order to highlight and review the impacts on processes and vice-versa. This allows for a stricter collaboration among the different stakeholders usually involved in BPR. These activities are mostly handled by the VLPM tool, as discussed below.

A. The VLPM Tool

VLPM [12] is a tool supported methodology for process modeling and re-engineering of PA, by providing a set of functions to synchronize models and XML representation of laws, thereby allowing traceability. The tool also supports an automatic generation of documentation in a human readable form (e.g., PDF or HTML), and of skeletons of law modifications based on the changes undergone by processes defined by the original law. The extended design of the framework makes the tool more flexible and functional in various areas. Among which we mention: support for different XML representation of laws, which are used by VLPM for linking process and laws; more flexibility in deployment — e.g., by allowing integration with freely available UML tools; integration with formal analysis techniques and methodologies for simulation and verification.

Figure 4 shows a high-level representation of the model elements, i.e. a meta-model for the VLPM tool. The diagram mainly shows the internal representation of the model elements. In the diagram, a process is realized as an observable activity executed by one or more actors. Actors can be extracted from the text of the law or can be defined manually. They are identified by means of an unambiguous identifier (extracted from the XML file containing the law information or manually specified) and a name. This could easily be extended in order to add more features. In the same way as actors, assets can be either extracted from the law or defined manually. If the assets are extracted from the law, we then store their initial states in the model and use our notation to define the changes that the assets undergo.

In addition to the modeling elements, we use a generic relationship elements to create specific sub-classes of relationship as shown in Figure 5. These are defined separately from the elements of the model. Actor-Actor relationships have different properties from Actor-Process relationships (e.g., the allowed stereotypes) and from Process-Process relationships. The association of a process with its executing actors is based on the static assignments of the responsibilities (set of roles, R) to the actors. This information can be extracted from the law or manually assigned after the actors have been identified. The use of an abstract relationship object allows us to create as many types of relationship as we need, with the only requirement of defining also a suitable translation of each

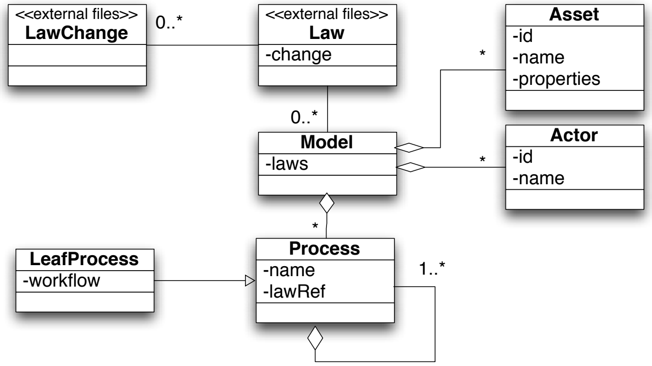

Fig. 4. The internal representation of our modeling elements.

relationship to UML. The model also explicitly support the Asset-Process relationships that define the semantics for the asset flows.

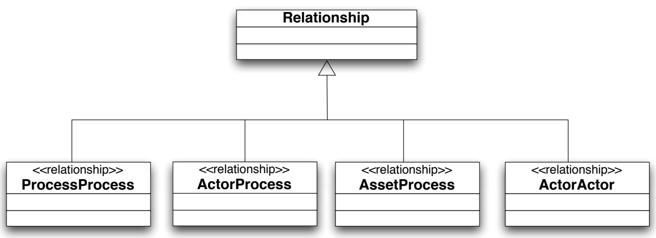

Fig. 5. Relationships among the modeling elements.

The model represents the static information of the business processes, while the dynamic properties (namely, asset transformation functions) are defined in a specific notation. The model is associated to the laws that regulate its business processes to allow the association of a single process with relevant law parts that define them. Notice, however, that the law is not included in the model. Although our model is designed to support XML format for laws representation, it can be easily extended to support other formats (see at the end of this section).

B. Methodology and Usage Scenario

We devised a methodology to automatically extract information from XML representation of laws and map them into process models. The core modeling elements are process, actor, asset, and relationships with triggering conditions. The methodology comprises of three steps.

The first step is the preparation of the data and structure of the model. Particularly, this step is responsible for identifying the actors, assets, stereotypes and terminologies, as well as responsible for collecting laws (the enumeration of laws which rule or influence the domain under analysis). The second step focuses on the use of UML *use case* diagrams to statically capture and analyze actors and processes independent from their execution. This is particularly important to breakdown processes hierarchically, to associate actors with responsibilities in the process breakdown structure, and to define and associate law paragraphs to processes in the use case diagram. Finally, the evolution of assets and processes are captured and analyzed using UML *activity diagrams*. Activity diagrams describe the processes workflow by emphasizing sequential and parallel activities (using the triggering conditions identified in step one) whose assets are needed and how their state evolve —i.e., how they change after being executing associated activities. The activity diagram also highlights the assets on which the processes operate. The connection between processes and assets are labelled with one or more of the following stereotypes: create, read, update, delete. In addition to the standard notation borrowed from the CRUD matrices [16], it is also possible to specify use, send and receive as stereotypes. This allows us to systematically translate into executable code (e.g., model checking) for further analysis, e.g., to perform procedural security analysis (such as, see in [21]).

The methodology also allows to link the laws and models. This particularly increases the traceability between laws and processes, with the goal of helping the law makers elaborate models in collaboration with software developers or process engineers, and understand the impact of law or process changes to their counterparts. This helps, first, to justify the existence of a particular process by providing a reference to the parts of the law that define it, which in turn allows us to link the process to all the constraints in the law that regulate it. Secondly, it allows to understand the impact of a change both in the law and in the process model. When a change is made to the law, on the one hand, being able to identify which processes are defined (or regulated) by the modified part of the law allows us to modify the process model accordingly. By looking at the model, it is then possible to determine what processes “interact” with the processes affected by the change in the law. The modification can then be propagated to all the relevant processes and makes the model up to date. On the other hand, the re-engineering of processes may result in a need to modify some parts of the law. Maintaining law-model traceability allows to automatically identify which parts of the law should be amended by tracing back to the parts of the law that originally defined the modified processes.

The points discussed above are supported by the VLPM tool, which has the following usage scenario (Figure 6 and see also in [12] for further detail):

1) A law written in natural language is marked with XML tags.
2) The user imports the law formatted in XML and VLPM generates a skeleton of the model. The user needs to verify and complete the generated model in order to have a reliable *as-is* view (i.e., a “process-tree” view) of the law. This model can be exported in various formats for documentation purposes (e.g., PDF).
3) The user imports an Explicit Text Amendment that modifies the law that has been previously modeled with VLPM. The tool highlights the impacts of the amendment on the law and on the model, allowing the user to focus on the affected parts of the model. This

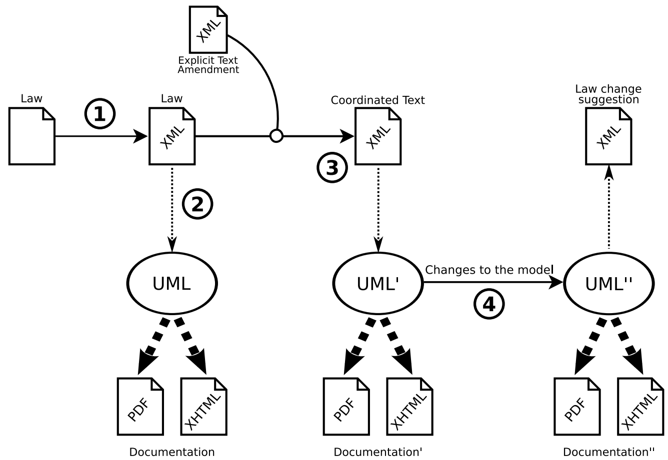

Fig. 6. Law modeling process handled by VLPM.

greatly simplifies the model revision process.

4) The user modifies the process model, re-engineering some processes. At this point documentation can be generated to be shared among the stakeholders and to compare the as-is and the to-be models. Moreover, VLPM can be used to generate the XML skeleton of a new law that amends the originally modeled law.

Furthermore, since the first version of VLPM focused on modeling the procedural aspects of legal documents that would allow to perform the necessary changes to the as-is business logic with the purpose. However, much information necessary for the reasoning cannot be easily extracted from the multitude of legal documents that regulate a domain without a background in jurisprudence. For this reason, the current version of VLPM (i.e., VLPM 2.0) exploits semantic markup on the laws to generate skeletons of business process models that can be traced back to the laws describing them. This allows lawyers and functional analysts to round-trip between laws and processes.

With VLPM 2.0, the components which produce the models from a set of legal documents provide some regulations for a certain domain. The information is layered hierarchically, where the bottom layer is the textual information and on top of which meta-data and structural information are added via the Akoma Ntoso [22], [23] markup XML format. We developed an OWL-DL ontology in order to add semantic information about processes described in legal texts, by extending the concepts of LKIF-core [24], [25] with a business process meta-model that borrows several entities from the BPMN meta-model [26]. We then defined some concepts that can effectively address our needs. We used Pellet Reasoner to identify and consolidate equivalences and other relations with LKIF-core classes. The VLPM 2.0 ontology is not a specification of the BPMN meta-model in OWL. Instead, it abstracts the core entities of a business process from the BPMN meta-model in order to obtain a smaller but more generic ontology. In the sense that a set of instances of the classes in such ontology can easily be transformed to BPMN as well as UML Activity Diagram (AD) [27]. Finally, we intend to support supplementary ontologies to allow the representation and modeling of other aspects of the domain, such as goal-oriented information.

V. Combining Nòmos and VLPM to Support BPR Scenario

Laws can express principles at different levels. Two levels are particularly apparent, high-level principles usually comprise of rules and requirements, and a set of procedural and/or operational level laws [28].

As we discussed previously, VLPM provides a robust environment to effectively manage the re-engineering of processes regulated by the set of operational laws. One significant limitation of the tool, however, is that it does not provide notations and means to represent the principles behind the procedures (or, better, motivating the procedures) and to reason about possible alternative implementation. Even from the business re-engineering point of view, such principles represent an essential part since they provide the framework and the constraints for the definition of new procedures and laws. This, in turn "moves" part of the re-engineering activity back to the "natural language" domain where inconsistencies and ambiguities might arise.

To overcome this problem, we propose the integration of goal-oriented methodology supported by Nòmos framework with the process modeling methodology supported by VLPM. The situation is depicted in Figure 7. On the left hand side, we have the law (possibly split in various documents) and typically describing general principles (e.g., "all citizens have the right to free health-care") and procedural and operational aspects (e.g., "to get free health-care you need a Social Security Number"). On the right-hand-side, we have to modeling techniques:

- Nòmos, in the upper part, provide a graphical notation and a methodology for modeling and reasoning about the high-level principles.
- VLPM, in the lower part, provide a graphical notation and methodology for modeling and reasoning about the procedures and lower-level operative principles.

Nòmos can represent the principles of the law via its constructs. In particular, as depicted in Figure 3(a) it is possible to represent the parts of a normative proposition such as the "Personal Health Information (PHI) closed", focusing on the actor, "Hospital", specified in the text of the law, also giving the possibility to specify the kind of right (in the case of the example a *duty*). Moreover, the framework allows to specify the goals that are induced by the text of the normative proposition, such as "set-up an encrypted electronic communication" and the specification of the actions that fulfill the goals and that represent the links to the procedural part of the law (in the example "Assign key to doctors and call center" and "Monitor electronic transaction"). Thanks to these representation, Nòmos maintains the complete knowledge about the principles at the bases of the operative part of the law, and of the possible alternatives for the law fulfilling (see Figures 3(b) and 3(c)).

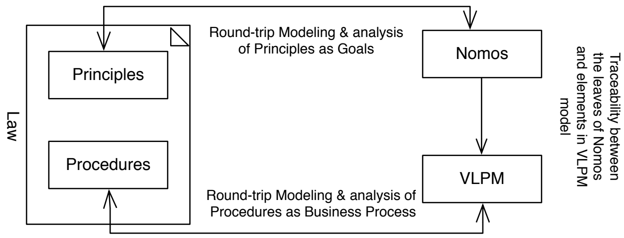

Fig. 7. The proposed approach for modeling and analysis of law.

VLPM extracts processes and actors from XML representation without the semantic knowledge that allows to reason on alternatives and here comes one of the essences of the Nòmos framework. Notice that the leaves of the Nòmos model can be analyzed on their fulfillment and on their compliance with the actual norm/law. There are also correlations between leaves in Nòmos and activities in VLPM. Thus, it is straightforward to say that such leaves can help enriching the VLPM "process-tree" —hierarchical decomposition of processes and actors using UML use cases— with more semantics on the management of activities in the VLPM model.

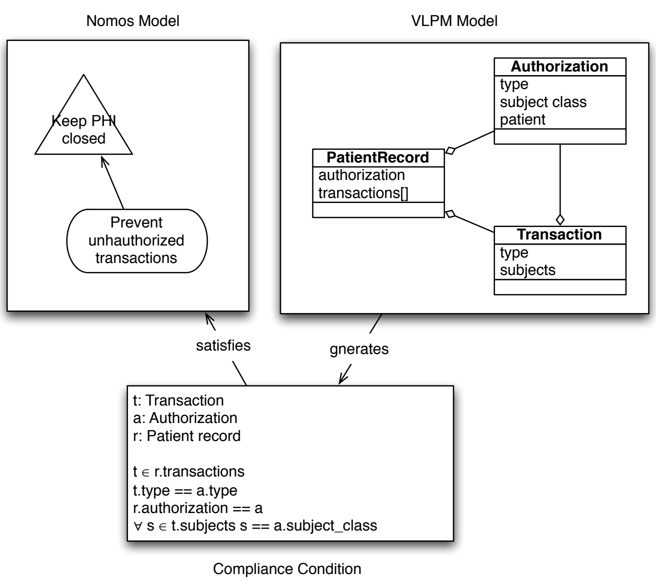

Fig. 8. A simple illustration using the proposed approach.

Figure 8 illustrates how the approach actually works. The Nòmos model contains the description of the principles that should be respected by the stakeholders. Moreover, it contains the description of the strategic goals that the stakeholders develop to be compliant. The presence of goals allows to make a domain-dependent analysis, which takes into account their specific objectives and needs, besides those of the law. The VLPM model contains a description of the system architecture. More precisely, it contains a description of that part of the system whose definition has been extracted from the annotated law. In the bottom part of the figure, the compliance condition is depicted. Basically, the compliance condition consists of a representation of the achievement condition of the goals of the Nòmos model. The condition says that the compliance goal "Prevent unauthorized transactions" is satisfied when the system is in a state represented by the value of the three variables t, a, and r, where t is an instance of the class `Transaction`, a is an instance of the class `Authorization`, and r is an instance of the class `PatientRecord`. The state represented by the condition is such that the transaction of certain data is linked with the patient's authorization to transmit that data. In the picture, this is considered true if:

- the transaction has also been associated to an authorization ($r.authorization == a$);
- the authorization is specific for that type of transaction ($t.type == a.type$);
- the transaction has actually been recorded into at least one patient record ($t \in r.transactions$) – i.e., no transaction happen, which are not registered and do not respect the other conditions;
- the subjects, among which the data is exchanged match the role type declared in the authorization for that transaction ($\forall s \in t.subjectss == a.subject\_class$).

Given this compliance condition between any possible state of the system, described in terms of the values of its variables and the principles expressed by the law, it is possible to exhaustively check for states that are allowed by the system but not acceptable for the law.

An important aspect to highlight is the traceability offered by the two approaches are complimentary. For example, if you decide to remove a process from the UML model that corresponds to one of the leaves of the Nòmos model, the Nòmos framework can trace up to the root goal and check if this action is complaint with the actual norm (from which the leave is derived). It is worth mentioning that when a new process is added to the model, VLPM generates a list of suggestions that can be used to produce an explicit text amendment from the changes undergone by the model, thus allowing the law to be realigned to the model. This can be further refined and enhanced by using the power of Nòmos analysis.

This conforms the connection between the Nòmos model and the VLPM model. The leaves of the Nòmos model, in fact, correspond to the procedures of the VLPM model. This provides a framework that allows to trace and reason either top-down, from the principles to the implementation, or, vice-versa, bottom-up, from a change in the procedure to the principles.

VI. Related Work

Several strategies have been proposed in the literature to understand, model, and analyze business process models. Three aspects are central in these approaches. The first is tools used for creating (business) process models. Second, notations used to represent the modeling elements and concepts. Third, techniques used for formally specifying and verifying how such models respect the intended goals.

With respect to modeling of business processes, for instance, various works in the past have been proposed for modeling business processes. These approaches span from workflow nets to event-based process chains, from flow-charts diagrams to UML Activity diagrams (ADs) and Business Process Execution Languages for Web Service (BPEL4WS)[29] and several other works such as [30], [31], [32]. In particular, [33], [34], [35] widely discussed the usage of UML ADs for modeling business processes as well as workflow modeling and specifications.

While these works demonstrated their usage scenarios for the modeling, specifying, and analyzing business processes and workflows of complex system, the attempt to model laws and procedures, as well as to perform formal analysis in favor of the public administration (PA) is not satisfactory.

In recent years, however, a number of governments have been adopted such techniques to support their PAs. Works like [3], [2], [36], [37] particularly have been discussed BPR support for public healthcare services by identifying different levels of process support and by distinguishing among generic process patterns. The use of BPR for better government has also been discussed earlier by the U.S. federal government and the U.S. Department of Defense and its use in taxation in [38]. The importance of modeling in the legal framework and documenting the knowledge about the legal constraints within the process model itself is stated in [4].

In particular, Olbrich and Simon in [39] discussed an approach based on event-driven process chains and suggested how to translate law paragraphs into process models using the Semantic Process Language (SPL). Their main goal is to the visualization and formal modeling of a legally regulated process. The interesting aspect of this work is not only the consideration of the given law when developing business process models, but also the explicit derivation of a process structure which is implicitly specified within the paragraphs themselves using the SPL. The SPL enabled them to articulate language structures into executable workflow models, using Petri Nets. The presented approach could provide means for verifying whether process-like behavior fulfills the selected paragraphs formally. Related to processes and their verification, in [40] the authors propose a UML-based approach to define, verify, and validate organizational processes, especially in the context of software process improvement and the CMMI (Capability Maturity Model Integration) framework.

Related to goal-oriented approaches for modeling and representation of laws and with the compliancy of set of requirements to laws. Three of them are particularly relevant for our work. In [8], the authors used KAOS as a modeling language for representing objectives extracted from regulation texts. Such an approach is based on the analogy between regulation documents and requirements documents. In [9], Goal-oriented Requirement Language (GRL) to model goals and actions prescribed by laws and exploit Use Case Maps (UCM) to describe the impact of laws on the business processes is discussed. This work is founded on the premise that the same modelling framework can be used for both regulations and requirements. In [28] is shown that two levels exist in legal systems: the Rule level, which gives prescriptions in an Event-Condition-Action (ECA) style; and the Requirements level, which expresses desirable state of affairs to be achieved by addressees. It also argues that the requirements level cannot exist alone: it depends on the rule level for actuation and enforcement. However, it tackles only with the requirements level while discussing the integration of laws into enterprise configurations.

Breaux et al. in [41] develop a systematic process called semantic parameterisation using the Cerno framework [42]. The approach consists of identifying in legal text restricted natural language statements (RNLSs) and then expressing them as semantic models of rights and obligations [10] (along with auxiliary concepts such as actors and constraints). Secure Tropos [43] is a framework for security-related goal-oriented requirements modeling that, in order to ensure access control, uses strategic dependencies refined with concepts such as: trust, delegation of a permission to fulfill a goal, execute a task or access a resource, as well as ownership of goals or other intentional elements. The main point of departure from Nòmos is that the Nòmos use a richer ontology for modeling legal concepts, adopted from the literature on Law. Additionally, the legal models one builds using Nòmos is different from the mentioned usage —i.e., Nòmos allows to check compliance between an i* model of system requirements and a model of a law fragment.

VII. Conclusion

The application of BPR and goal-oriented in law modeling and analysis can facilitate the work of PAs by favoring the involvement of citizens in (the law) decision process. The definition of strict constraints for the structure of a law facilitates its readability and editing, but —in the case of laws definition procedures —the use of (visual) representations, their modeling, and formal reasoning can take this even further.

This paper proposed an approach intended to provide systematic support for modeling and analyzing laws. Our approach combines two existing complementary frameworks that tackle the discussed concerns in different levels of abstractions. While one (i.e., Nòmos) exploits goal-oriented approach, the other (i.e., VLPM) focuses on the use of UML-based BPR approach. We emphasized on the integration of these two approaches for realizing principles, procedures, as well as operational aspects of the law, and for developing a system that can maintain and support the laws.

The resulting analysis method of the Nòmos approach takes advantage of two key ideas, namely the concept of intentional compliance to verify law-compliance of requirements models, and the idea of combining a law model with an intentional model of requirements for preserving the explicit representation of compliant alternatives resulting from goal analysis. In contrast, VLPM is based on well established technologies for legal knowledge representation such as LKIF and Akoma Ntoso, including process and goal-oriented ontologies. The current VLPM framework will have the possibility to play an important role in the "ICT for law" initiatives and eventually become an actual tool for the improvement of Public Administrations.

In summary, the proposed approach has given promising clues to trace and reason laws either from the principles to the implementation, or, vice-versa, from a change in the procedure to the principles. Although, the snippet example used for our proof-of-concept is not complete, we believe that the proposed approach can be refined and (re-)used to model and analyze different laws, as long as the laws describe both the principles and procedures. Namely, as long as the the laws define, regulate or in some way affect procedures, e.g., PA procedures, company policies that need to comply with certain regulations. Moreover, the implementation of the approach in terms of a tool is not discussed. However, we are currently working on enriching the two frameworks because this would allow us to develop a machinery for the combined activities. We are also looking for a real case study to evaluate our approach.

REFERENCES

[1] A. Villafiorita, K. Weldemariam, A. Susi, and A. Siena, "Modeling and Analysis of Laws Using BPR and Goal-Oriented Framework," *International Conference on the Digital Society*, pp. 353–358, 2010.

[2] W. C. Leslie Willcocks and S. Jackson, "Radical Re-Engineering and Information Systems: Evidence from UK Public Services," in *Fifth European Conference in Information Systems*. Cork, 1997.

[3] V. B. Marcel Thaens and H. van Duivenboden, "Business Process Redesign and Public Administration: a Perfect Match?" in *Taylor, J.A., Snellen, I.Th.M. and Zuurmond, A. (Eds.): Beyond BPR in Public Administration: An Institutional Transformation in an Information Age*. IOS Press, Amsterdam, 1997, pp. 15–36.

[4] P. Alpar and S. Olbrich, "Legal Requirements and Modelling of Processes in e-Government," *Electronic Journal of e-Government*, vol. 3, 2005.

[5] A. Siena, "Engineering Law-Compliant Requirements. The Nòmos Framework." Ph.D. dissertation, Department of Information Engineering and Computer Science, University of Trento, Trento, March 2010. [Online]. Available: http://eprints-phd.biblio.unitn.it/230/

[6] R. Kowalski and M. Sergot, "Computer Representation of the Law," in *Proceedings of the 9th international joint conference on Artificial intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1985, pp. 1269–1270.

[7] T. J. M. Bench-Capon, G. O. Robinson, T. W. Routen, and M. J. Sergot, "Logic Programming for Large Scale Applications in Law: A Formalisation of Supplementary Benefit Legislation," in *Proceedings of the 1st international conference on Artificial intelligence and law*, ser. ICAIL '87. New York, NY, USA: ACM, 1987, pp. 190–198.

[8] R. Darimont and M. Lemoine, "Goal-oriented Analysis of Regulations," in *ReMo2V*. CEUR-WS.org, 2006.

[9] S. Ghanavati, D. Amyot, and L. Peyton, "Towards a Framework for Tracking Legal Compliance in Healthcare," in *Proceedings of the 19th international conference on Advanced information systems engineering*, ser. CAiSE'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 218–232.

[10] T. D. Breaux, A. I. Antón, and J. Doyle, "Semantic Parameterization: A Process for Modeling Domain Descriptions," *ACM Trans. Softw. Eng. Methodol.*, vol. 18, no. 2, pp. 1–27, 2008.

[11] A. Siena, J. Mylopoulos, A. Perini, and A. Susi, "A Meta-Model for Modeling Law-Compliant Requirements," in *Proceedings of RELAW workshop at RE 2009*. IEEE Computer Society, 2009.

[12] A. Ciaghi, A. Mattioli, and A. Villafiorita, "VLPM: A Tool to Support BPR in Public Administration." in *ICDS*. IEEE Computer Society, 2009, pp. 289–293.

[13] A. Ciaghi, A. Villafiorita, and A. Mattioli, "A Tool Supported Methodology for BPR in Public Administrations," *International Journal of Electronic Governance*, vol. 3, no. 2, 2010.

[14] R. Hoekstra, J. Breuker, M. Di Bello, and A. Boer, "LKIF Core: Principled Ontology Development for the Legal Domain," in *Proceeding of the 2009 conference on Law, Ontologies and the Semantic Web*. Amsterdam, The Netherlands: IOS Press, 2009, pp. 21–52.

[15] L. Sommerville, *Software engineering (5th ed.)*. Addison Wesley Longman Publishing Co., Inc., 1995.

[16] A. Mattioli, "Analysis of Processes in the Context of Electronic Election," Master's thesis, University of Trento, Italy, 2006, in Italian.

[17] L. Wenar, "Rights," in *The Stanford Encyclopedia of Philosophy*, fall 2010 ed., E. N. Zalta, Ed., 2010.

[18] W. N. Hohfeld, "Some Fundamental Legal Conceptions as Applied in Judicial Reasoning," *Yale Law Journal*, vol. 23, no. 1, 1913.

[19] E. S.-K. Yu, "Modelling Strategic Relationships for Process Reengineering," Ph.D. dissertation, University of Toronto, Toronto, Ontario, Canada, 1996.

[20] A. Susi, A. Perini, J. Mylopoulos, and P. Giorgini, "The Tropos Metamodel and Its Use." *INFORMATICA*, vol. 29, no. 4, pp. 401–408, 2005.

[21] K. Weldemariam and A. Villafiorita, "Formal Procedural Security Modeling and Analysis," in *International Conference on Risks and Security of Internet and Systems*, ser. CRiSIS '08. Washington, DC, USA: IEEE, October 2008, pp. 249–254.

[22] United Nations Department of Economic and Social Affairs. (2010) Akoma Ntoso Framework: Architecture for Knowledge-Oriented Management Of African Normative Texts Using Open Standards and Ontologies. [Online]. Available: http://www.akomantoso.org/

[23] F. Vitali and F. Zeni, "Towards a Country-Independent Data format: the Akoma Ntoso Experience." in *Proceeding of the V Legislative XML Workshop*, 2007, pp. 239–252.

[24] ESTRELLA project. (2010) LKIF-Core Ontology: A Core Ontology of Basic Legal Concepts. [Online]. Available: http://www.estrellaproject.org/lkif-core/

[25] R. Hoekstra, J. Breuker, M. D. Bello, and A. Boer, "The LKIF Core Ontology of Basic Legal Concepts," in *Proceedings of the Workshop on Legal Ontologies and Artificial Intelligence Techniques (LOAIT 2007)*, P. Casanovas, M. A. Biasiotti, E. Francesconi, and M. T. Sagri, Eds., June 2007.

[26] "Business Process Modeling Notation (BPMN) Version 1.2," January 2009. [Online]. Available: http://www.omg.org/spec/BPMN/1.2/PDF

[27] G. Booch, J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional, 2005.

[28] W. Hassan and L. Logrippo, "Requirements and Compliance in Legal Systems: A Logic Approach," *Requirements Engineering and Law*, pp. 40–44, 2008.

[29] M. B. Juric, *Business Process Execution Language for Web Services BPEL and BPEL4WS 2nd Edition*. Packt Publishing, 2006.

[30] J. Lee, "Goal-Based Process Analysis: A Method for Systematic Process Redesign," in *Proceedings of the Conference on Organizational Computing Systems*, ser. COOCS '93. New York, NY, USA: ACM, 1993, pp. 196–201.

[31] M. M. Lehman, "Process Modeling —Where Next," in *Proceedings of the 19th Internaional conference on Software Engineering*, ser. ICSE '97. New York, NY, USA: ACM, 1997, pp. 549–552.

[32] V. Hlupic, "Business Process Modelling Using Discrete Event Simulation: Potential Benefits And Obstacles For Wider Use," *International Journal of Simulation: Systems, Science and Technology*, vol. 7, pp. 62–67, 2003.

[33] M. Dumas and A. H. M. t. Hofstede, "UML Activity Diagrams as a Workflow Specification Language," in *Proceedings of the 4th International Conference on The Unified Modeling Language, Modeling Languages, Concepts, and Tools*. London, UK: Springer-Verlag, 2001, pp. 76–90.

[34] N. Castela, J. M. Tribolet, A. Silva, and A. Guerra, "Business Process Modeling with UML," in *ICEIS (2)*, 2001, pp. 679–685.

[35] R. Eshuis, "Symbolic Model Checking of UML Activity Diagrams," *ACM Trans. Softw. Eng. Methodol.*, vol. 15, no. 1, pp. 1–38, 2006.

[36] G. Greco, A. Guzzo, and L. Pontieri, "Mining Hierarchies of Models: From Abstract Views to Concrete Specifications," in *Business Process Management*, 2005, pp. 32–47.

[37] R. Lenz and M. Reichert, "IT Support for Healthcare Processes," in *Business Process Management*, 2005, pp. 354–363.

[38] U.S.A. National Performance Review, "Executive Summary — Creating a Government that Works Better and Costs Less," 1993. [Online]. Available: http://govinfo.library.unt.edu/npr/library/nprrpt/annrpt/redtpe93

[39] S. Olbrich and C. Simon, "Process Modelling towards e-Government – Visualisation and Semantic Modelling of Legal Regulations as Executable Process Sets," *Electronic Journal of e-Government*, vol. 6, 2008.

[40] N.-L. Hsueha, W.-H. Shen, Z.-W. Yanga, and D.-L. Yanga, "Applying UML and software simulation for process definition, verification, and validation," *Information and Software Technology*, vol. 50, pp. 897–911, 2008.

[41] T. D. Breaux, M. W. Vail, and A. I. Anton, "Towards Regulatory Compliance: Extracting Rights and Obligations to Align Requirements with Regulations," in *RE'06: Proceedings of the 14th IEEE International Requirements Engineering Conference*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 46–55.

[42] N. Kiyavitskaya, N. Zeni, J. R. Cordy, L. Mich, and J. Mylopoulos, "Cerno: Light-Weight Tool Support for Semantic Annotation of Textual Documents," *Data Knowl. Eng.*, vol. 68, no. 12, pp. 1470–1492, 2009.

[43] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "Requirements Engineering meets Trust Management: Model, Methodology, and Reasoning," in *ITRUST-04*, ser. LNCS, vol. 2995. SVG, 2004, pp. 176–190.