Modeling and Analysis of Laws using BPR and Goal-Oriented Framework

Adolfo Villafiorita, Komminist Weldemariam, Angelo Susi, Alberto Siena
Center For Information Technology
Fondazione Bruno Kessler-irst
Trento 38100, Italy
Email: (adolfo,sisai,susi,siena)@fbk.eu

***Abstract*—Recently, two complementary approaches are proposed to represent, model, and analyze laws: the Nomos and VLPM approaches. Nomos is a goal-oriented approach to effectively capture high-level principles in terms of goal realization for requirements guided by satisfiability of normative propositions obtained from rules embedded in a law. The latter offers a tool supported (re-)engineering methodology to extract laws represented in XML and build models using a subset of UML diagrams. Both allow traceability between laws and their respective models.**

This paper proposes an integration of these two approaches. We believe that this provides a framework that allows to trace and reason either top-down, from principles to their implementation or, viceversa, bottom-up, from a change in the procedure to the principles. It is exactly this connection that adds value to the solution we propose and makes our approach more significant than a simple juxtaposition of the two techniques.

***Keywords*-BPR, Laws, Goal-Oriented, Nomos, VLPM.**

I. Introduction

New technologies and the emergence of new paradigms in the relationship between Citizens and Governments are constantly challenging and questioning the way in which Public Administration (PA) delivers its services. This phenomenon is as old as Governments are: the Roman Emperor Augustus had among his most relevant reforms an improvement of the PA [1].

Traditionally, there are three elements on which Governments can operate to improve the way in which they deliver services: people, processes, and tools. In Augustus' case, the reform was implemented through people: he transitioned from short-term "amateurs" taken from the privileged class to "paid professionalized civil servants" [1]. In recent times, interventions always encompass all three dimensions. We mention, for instance, [2] and [3].

Not surprisingly, what makes the problem even more challenging today is, in our opinion, the pace at which opportunities to innovate and improve arise. Governments, however, cannot afford mistakes and changes that require careful cost-benefit analyses. The introduction of new systems and procedures, moreover, requires a careful analyses of laws to ensure no conflict arises between the way things are done (processes) and the way things are meant (laws). Therefore, it is very important to support the correct implementation and execution of these concerns, as they are of the essence for public administrations.

One way to do this is, by using tools and techniques to specify, model, and analyze the underlying concepts of the laws —i.e., principles and procedures. In fact, modeling and reasoning of laws using a computer system is not new [4], [5]. And following these initial works, various approaches have been proposed (see, e.g., [6], [7], [8], [9]).

Previously, we proposed two approaches with a particular emphasis on supporting PAs. We did so, by presenting a methodology based on goal-oriented analysis —the Nomos framework [10] and based on process re-engineering using UML diagrams —the VLPM framework [11]. The two approaches are supported by respective tools to allow traceability between laws and models. However, the two approaches individually have significant limitations. For example, VLPM does not provide notations and means to represent the principles behind the procedure and means to reason about possible alternative implementation. In contrast, the *Nomos* approach does not provide low-level implementation of the actual processes impeding the analysis of some components of the law.

The focus of this paper is, therefore, presenting some of our efforts to help PAs in managing and supporting changes by combining these two complimentary approaches. We believe, in fact, that these components are important building blocks to help shortening adoption times, while, at the same time, providing structured ways to reason about the costs and benefits of different ways in which new technologies and new paradigms can be introduced.

This next section presents the business process re-engineering context (BPR) for modeling and analysis of public administrations. Section III discusses how such processes can be realized at higher-level using goal-oriented framework. Section IV discusses the VLPM approach for supporting PAs. We discuss our attempt to support BPR using the best out-of these two techniques in Section V. Finally, we present some related work, and conclusion and possible future directions in Sections VI and VII.

II. Motivation: BPR Context

(Business) Process Re-engineering (BPR) helps an organization to better achieve its strategic goals by acting

978-0-7695-3953-9/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDS.2010.57

on three dimensions: people (and their skills), technology, and processes. At a very high level and with several simplifications, BPR consists of four main steps: modeling the current processes (the as-is), defining goals of the re-engineering activity (e.g., increase efficiency, reduce costs, introduce a new function), devising new processes (the to-be), and actuate them. Both the as-is and to-be can be represented in some textual or graphical representations. In the implementation, the as-is situation (e.g., actual skills, processes, and systems) is the starting point that might limit the scope of intervention.

When we look at process re-engineering in addition to the constraints mentioned above, we need to take into account also the law that regulates the goals, missions, and often, also the way in which work has to be organized.

In principle, thus, we distinguish three different kinds of re-engineering projects:

- **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 laws.
- **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.
- **Inter-departmental level.** The goal of this kind of projects 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. We will not consider it anymore in the rest of this paper.

The other two kind of projects, however, present two peculiar and closely related challenges, whose root is in the relationship between the laws and the processes that implement such laws. The challenges are,

- **Challenge 1.** Laws provide the framework that constrains and limits possible choices and alternatives in re-engineering processes. Providing tools and notations allowing us to explicitly reason about such alternatives and constraints, and helping us to develop more efficient solutions.
- **Challenge 2.** 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 helps to ensure that the procedure correctly implement the law and, at the same time, it helps to understand which laws might be affected by a change in the processes.

We believe that, first, goal-oriented methodologies help tackling the first problem by providing precise notations to avoid mis-interpretation and resolve ambiguities that can arise, and by performing high-level formal reasoning. The Nomos framework is one of these approaches. Second, we believe that the VLPM —a tool we developed for the purpose, helps tackling the second challenge. In the rest of the paper, therefore, we start by briefly presenting these methodologies and we then provide a proof of concept of the advantages we can get by putting the two together.

III. Goal-oriented Approach for Law Compliance

In business process (re)engineering and in requirements engineering, a preliminary objective consists of having the comprehension of the "as-is" as the key to understand how the new system or revised process has to be structured. In this context, goal analysis techniques have been proposed in the last decade to understand the structure and the correlations of stakeholders goals, sub-goals, and their relation with operational plans. This allows to answer the *why* questions, besides the *what* and *how*, regarding system functionality.

For example, *i** [12] is a goal-oriented modeling framework. The framework is conceived as a conceptual tool for business process (re)engineering, in particular, to support the exploration of the as-is of the domain. It uses primitive concepts, such as actor, goal, task, resource, and dependencies among the elements for modeling requirements. The framework represents a domain as a network of actors, who depend on each other to achieve goals or perform tasks. Goals represent stakeholders needs, and can be AND- or OR- refined into sub-goals, or linked with each other through contribution links. Goal refinement continues until goals can be operationalized by means of tasks —that is, concrete actions that can be performed by actors. From the goals' operationalization, the concrete requirements specification for the system-to-be can then be derived. Alternative goal refinements, together with goals operationalization, allow to reason about possible alternative requirements. (See in [12], for a detailed description of the *i** framework.)

This way, goal-orientation results in linking system functionalities to stakeholder needs, preferences and objectives, and to explore alternatives through goal reasoning techniques.

Nomos [10] is a modeling framework that extends the $i*$ language for arguing about compliance of requirements. It moves from the consideration that requirements should not be aligned only with stakeholders needs. In many cases —specially when dealing with the public administrations— laws are a rich source for requirements that must be aligned with the originating law, or can cause severe legal consequences if not modeled and aligned with the law in a correct way. Nomos relies on the Hohfeldian taxonomy of legal concepts to enrich the $i*$ language. Duties, rights, privileges, liabilities, and so on are modeled into the domain model and assigned to the addressed stakeholders. Afterwards, the proper strategy to comply with the modeled law is searched, by adding goals to the model. Further refinement of a goal into sub-goals and tasks allows to operationalize legal prescriptions. Alternatives may be explored by looking at the opportunities given by the law and by conceiving different ways to fulfill a given legal prescription. This way, different operationalizations of the law can be analyzed.

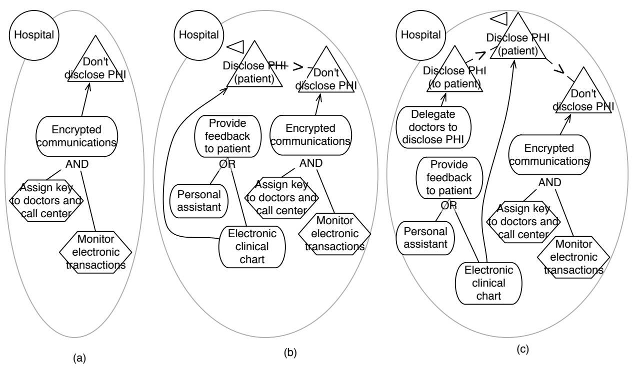

Figure 1. The Nomos modelling language.

Figure 1(a) depicts excerpt model of a law fragment taken form the U.S. Health Insurance Portability and Accountability Act (HIPAA). 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 1(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 1(c), the introduction of the last principle —a duty, for the Hospital, to disclose such information to the patient upon request— further impacts the supported processes, as it requires the hospital to receive the requests and to delegate somebody to disclose the data.

IV. Enhanced VLPM Scenario for PAs

VLPM [11] is a tool supported methodology for process modeling and re-engineering of public administration. It provides a set of functions to synchronize models and XML representation of laws —namely, it provides traceability between the laws and models. The tool supports an automatic generation of documentation in human readable form (e.g., PDF or HTML). Moreover, it allows to automatically generate skeletons of law modifications based on the changes undergone by processes defined by the original law. We also extended the tool to make it 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 law); 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 [13], [14], [15]).

A. 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, the 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 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 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 execution of activities. The activity diagram also highlights the assets in which the processes operate. The connection between processes and assets are labelled with one or more of the following stereotypes: create, read, update, and delete. In addition to the standard notation borrowed from the CRUD matrices, 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 [13]).

Using the methodology, it is straightforward to link the laws and models in order to increase the traceability between them. One of the goals of traceability is helping law makers elaborate models in collaboration with software developers or process engineers, and understand the impact of law or process changes. 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, 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 2 and see also in [11] for further detail):

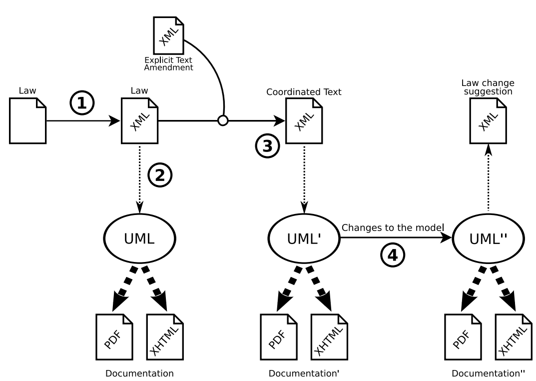

Figure 2. Law modelling process handled by VLPM.

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;
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 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.

V. Supporting BPR Scenario

Laws can express principles at different levels [16]. Two levels are particularly apparent, high-level principles usually comprise of rules and requirements, and a set of procedural or operational level or both laws.

As we have shown in the previous section, VLPM provides a robust environment to efficiently 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 in which inconsistencies and ambiguities might arise.

To overcome this problem, we propose the integration of goal-oriented methodologies supported by Nomos framework with the process modeling methodologies supported by VLPM. The situation is depicted in Figure 3. 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 SSN"). On the right-hand-side, we have two modeling techniques:

- Nomos, 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 a methodology for modeling and reasoning about the procedures and lower-level operative principles.

On its side, *Nomos* can represent the principles of the law through its constructs. In particular, as depicted in Figure 1(a) it is possible to represent the parts of a normative proposition (such as, the "Personal Health Information (PHI) closed") by focusing on the actor (e.g., "Hospital") specified in the text of the law, also giving the possibility to specify the kind of right (in the example, a *duty*). Moreover, the framework allows to specify goals that are induced by

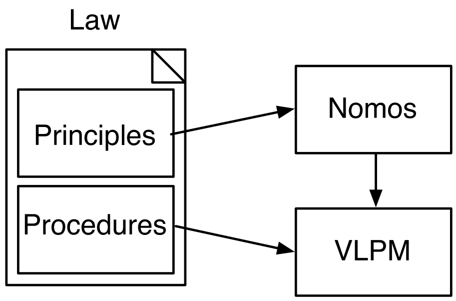

Figure 3. The approach

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”). Nomos 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 fulfillment, as depicted in Figures 1(b) and 1(c).

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 *Nomos* framework. Notice that the leaves of the *Nomos* model can be analyzed on their fulfillment and on their compliance with the actual norm or law. There are also correlations between leaves in Nomos and activities in VLPM. Thus, it is straightforward to say that such leaves can help enriching the VLPM “process-tree” with more semantics on the management of activities in the VLPM model.

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 *Nomos* model, the *Nomos* 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).

As noted previously, 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 *Nomos* analysis. This confirms the connection between the *Nomos* model and the VLPM model. The leaves of the Nomos 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

Various works describe how to use modeling languages and formal methods for modeling, specifying, and analyzing business processes and workflows can be found, e.g., [17], [18], [19]. However, little is usually said on the attempt to model laws and procedures, and perform formal analysis in favor of the public administration processes. In [8], [9], [20], [21], the authors discussed advantages and difficulties related to the (re-)engineering of public services. Works like [22], [23] particularly discuss ICT support for public *healthcare* services by identify different levels of process support and distinguish between generic process patterns.

Related to processes and their verification, in [24] 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 framework. The importance of modeling in the legal framework and documenting the knowledge about the legal constraints within the process model itself is stated in [9], [25]. The authors propose an approach based on Event-driven process chains and suggest how to translate law paragraphs into process models using the Semantic Process Language (SPL).

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 [6], the authors used KAOS as a modelling language for representing objectives extracted from regulation texts. Such an approach is based on the analogy between regulation documents and requirements documents.

In [7], 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 modeling framework can be used for both regulations and requirements. In [16] 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 can’t exist alone: it depends on the rule level for actuation and enforcement. However, it tackles only the requirements level while discussing the integration of laws into enterprise configurations.

VII. Conclusion and Future Work

We argue that there is a need for harmonization between (national) laws, by revising them or possibly introducing new ones, in order to achieve a new era of eDemocracy. This cannot be achieved without correct implementation and execution of the laws, which require to be correctly represented and modeled, and formally validated.

In fact, the application of BPR and goal-oriented in law modeling and analysis can facilitate the work of public administrations and favor 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.

Although the approach we discussed needs to be further elaborated and refined, in this paper we highlighted some of our efforts in tackling the mentioned challenges. Two complimentary approaches, the goal-oriented and UML-based, are described. We then 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.

Some of our future directions include developing a machinery to automate the approach we discussed and devising a precise methodology to be used along with this machinery. We are also looking for a case study to evaluate our approach.

REFERENCES

[1] F. Heady, *Public administration : a Comparative Perspective*, Fourth edition ed., ser. Public administration and public policy ; 41, 1991.

[2] U. N. P. Review, "Executive Summary — Creating a Government that Works Better and Costs Less," 1993.

[3] P. Shvaiko, A. Villafiorita, A. Zorer, L. Chemane, T. Fumo, and J. Hinkkanen, "eGIF4M: eGovernment Interoperability Framework for Mozambique," in *EGOV*, 2009, pp. 328–340.

[4] R. A. Kowalski and M. J. Sergot, "Computer Representation of the Law," in *IJCAI*, 1985, pp. 1269–1270.

[5] T. J. M. Bench-Capon, G. O. Robinson, T. Routen, and M. J. Sergot, "Logic Programming for Large Scale Applications in Law: A Formalisation of Supplementary Benefit Legislation," in *ICAIL*, 1987, pp. 190–198.

[6] R. Darimont and M. Lemoine, "Goal-Oriented Analysis of Regulations," in *ReMo2V, held at CAiSE'06*, 2006.

[7] S. Ghanavati, D. Amyot, and L. Peyton, "Towards a Framework for Tracking Legal Compliance in Healthcare." in *CAiSE*, ser. Lecture Notes in Computer Science, J. Krogstie, A. L. Opdahl, and G. Sindre, Eds., vol. 4495. Springer, 2007, pp. 218–232.

[8] Marcel Thaens, Victor Bekkers and Hein 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.

[9] Paul Alpar, Sebastian Olbrich, "Legal Requirements and Modelling of Processes in e-Government," *Electronic Journal of e-Government*, vol. 3, 2005.

[10] 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*, 2009.

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

[12] E. S.-K. Yu, "Modelling Strategic Relationships for Process Reengineering," Ph.D. dissertation, Toronto, Ont., Canada, Canada, 1996.

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

[14] A. Xenakis and A. Macintosh, "Procedural Security and Social Acceptance in E-Voting," in *HICSS '05: Proceedings of the Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS'05) - Track 5*. Washington, DC, USA: IEEE Computer Society, 2005, p. 118.1.

[15] K. Weldemariam and A. Villafiorita, "Modeling and Analysis of Procedural Security in (e)Voting: The Trentino's Approach and Experiences," in *EVT*, 2008.

[16] W. Hassan and L. Logrippo, "Requirements and compliance in legal systems: a logic approach," *Requirements Engineering and Law*, vol. 0, pp. 40–44, 2008.

[17] V. S. W. Lam and J. Padget, "Symbolic Model Checking of UML Statechart Diagrams with an Integrated Approach," *Engineering of Computer-Based Systems, IEEE International Conference on the*, vol. 0, p. 337, 2004.

[18] F. Puhlmann and M. Weske, "Using the *pi*-Calculus for Formalizing Workflow Patterns," in *Business Process Management*, 2005, pp. 153–168.

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

[20] David G. Wastell, P. White, P. Kawalek, "A Methodology for Business Process Redesign: Experiences and Issues," *Journal of Strategic Information Systems*, vol. 3, pp. 23–40, 1994.

[21] Alexandros Xenakis and Ann Macintosh, "Using Business Process Re-engineering (BPR) for the Effective Administration of Electronic Voting," *The Electronic Journal of e-Government*, vol. 3(2), 2005.

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

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

[24] Nien-Lin Hsueha, Wen-Hsiang Shen, Zhi-Wei Yanga, and Don-Lin Yanga, "Applying UML and software simulation for process definition, verification, and validation," *Information and Software Technology*, vol. 50, pp. 897–911, 2008.

[25] Sebastian Olbrich, Carlo 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.