Supporting Public Administration with an Integrated BPR Environment

Aaron Ciaghi[1], Adolfo Villafiorita[1], Komminist Weldemariam[1,2], Andrea Mattioli[1], and Quoc-Sang Phan[1,2]

[1] Fondazione Bruno Kessler (fbk-irst), Italy
via Sommarive 14, I-38050 Trento, Italy
{ciaghi,adolfo,sisai,amattioli,phan}@fbk.eu

[2] Department of Information Engineering and Computer Science,
University of Trento, via Sommarive 18, I-38050 Trento, Italy
{weldemar@disi,quocsang.phan@studenti}.unitn.it

Abstract. The definition or redesign of Public Administration (PA) procedures is particularly challenging. This is, for example, due to the requirement of cooperation of different organizational units and actors, different laws and procedures for the production of several artifacts, and maintaining traceability while integrating processes with new laws.

We are interested in business process modeling and re-engineering (BPR) for PA, where ICT can play a pivotal role by, e.g., improving communication among law-makers and process analysts. With regard to this previously we developed a tool called VLPM[1]. The tool is designed to provide assistance in BPR for PA, which allows traceability between laws and processes. In this paper, we discuss the extension of the tool and of its methodology to support an integrated environment that can be used for better law and process re-design by performing formal analysis on the processes. We discuss its system components and provide a working example taken from the Italian Immigration law, as a proof of concept.

1 Introduction

One of the most recent trends undertaken by the ICT community in the support of public administration (PA) is based on the use of business process modeling (BPR) techniques [1,2,3]. More specifically, the use of modeling languages to graphically represent PA procedures as business processes in order to redesign such procedures is becoming a widespread practice. For instance, the use of BPR for better government has been discussed earlier by the US federal government and the US Department of Defense [4]. Its specific uses for public services such as in taxation and in healthcare, widely discussed, e.g., in [5,6,7,8]. This approach is helpful not only in a process redesign scenario but also in situations where legal procedures are neither well-documented or clearly-defined nor presented at all (e.g., think of the land management in some developing countries).

[1] Visual Law Process Modeler: http://ed.fbk.eu/vlpm/

A. Villafiorita, R. Saint-Paul, and A. Zorer (Eds.): AFRICOMM 2009, LNICST 38, pp. 50–59, 2010.

Various works that describe how to use modeling languages and formal methods for modeling, specifying and analyzing business processes and workflows have been presented in, e.g., [9,10]. However, little is usually said on the attempt to model laws and procedures, and perform formal analysis in favor of the PA processes. To tackle this, in [11] we presented a tool called VLPM. The tool uses a subset of UML diagrams guided by strictly defined methodology in order to represent the laws and procedures as business process models, which are organized as static and dynamic process views and support their manual analysis. It also generates a well-formed documentation (e.g., in HTML or PDF).

In this paper, we discuss the roadmap for VLPM and a set of first extensions aimed at providing a new set of functions for analyzing, while, at the same time, improving interaction, usability, and portability.

2 Challenges of ICT-Supported PA

Procedures in PA are typically encoded in formal documents and laws. In what follows we discuss the main challenges we experienced in representing, modeling, and analyzing legal documents as (business) process. These challenges, in fact, represent a set of user requirements for the development of VLPM$^+$.

1. **Interpretation.** In several countries information related to legal regulations are spread across laws and procedures. This can allow different actors to interpret the same concept encoded in the same law or procedure differently. As a result, this arises issues related to *inconsistencies*, when the definition of one law conflicts with other law(s). For instance, an Italian old but still valid royal charter states t hat only who is Italian or European citizen can be employed in public transportation services. However, another law promulgated in 1998 states that any resident in Italy can be employed in the public service.
2. **Methodology and Tools.** Using modeling tools (such as, UML Activity Diagrams [12], BPML [13]) for creating workflow (or activity) diagrams has different advantages. For example, the mapping and visualization of the processes help the stakeholders to easily communicate and understand the processes. However, as pointed out also in [2], customizations are needed to make the representation clear and usable.
3. **Change Management Support.** When process modeling is used with the goal of re-engineering processes, an important aspect is guaranteeing "synchronization" between the model representation and the laws, so that any change to the process can be reflected in an equivalent amendment to the law and the other way round.

Besides the above challenges we also mention some of requirements for modeling e-Government processes given in [2]:

1. process models must contain the relevant subjects, objects, activities, events and constraints of administrative processes that make up a transaction;

2. e-Government process models should be standardized so that they can be synchronized and put together with other such processes to form a one-stop solution for their end users;
3. the resulting models shall be able to show the restrictions for re-engineering that are set by the legal framework or other public regulations;
4. method and notation must not be too complex since administrative executives are usually not familiar with modeling languages.

3 Framework for Supporting PA

VLPM is a tool, freely available tool and built as a Visual-Paradigm plug-in, designed to support process modeling and re-engineering for the public administration. It provides a set of functions to synchronize models and XML representation of laws. It is worth mentioning that as of the currently version the tool only supports the Italian laws. We are working to make the tool more flexible and more 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 (for simulation and verification, e.g., [14].

The resulting tool, VLPM$^+$, will provide the following functions:

1. *Provide static view of processes.* It consists of of a set of modeling methodology and functions to help in identifying, representing, and modeling laws and procedures as business processes. (Currently provided by VLPM.)
2. *Change Management and Traceability of the Process.* Maintaining law-model traceability while changing the model can allow 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. Performing re-engineering on some process might also require to update the model, which in turn must ensure traceability between laws and new processes. (Partly provided by VLPM.)
3. *Generate Documentation.* To improve the information sharing and further communication among the different organizational units as well as actors VLPM generate a well-formed documentation. (Currently provided by VLPM).
4. *Formal Analysis of the Model.* Here, our aim is to support analysis of the processes with formal methods (such as, model checking and simulation). In this way we can formally analyze laws and procedures, define a boundary to perform procedural security analysis, and list suggestions and identify critical points. (New function of VLPM$^+$).
5. *Customization and Installation Options.* VLPM is based on two standards, the Italian XML representation of laws and the UML. VLPM+ will support customization of the XML standard to support schema adopted by other nations and allow for the integration of UML tools providing standard connection “ports” to the UML (e.g., XMI).

4 System Description

4.1 Intermediate Representation of Model Elements

In order to have a representation of processes flexible enough to represent the portability and usability scenarios described above and support formal analysis, we developed the metamodel shown in Figure 1. The core modeling elements are *process, actor, asset*, and *relationships*.

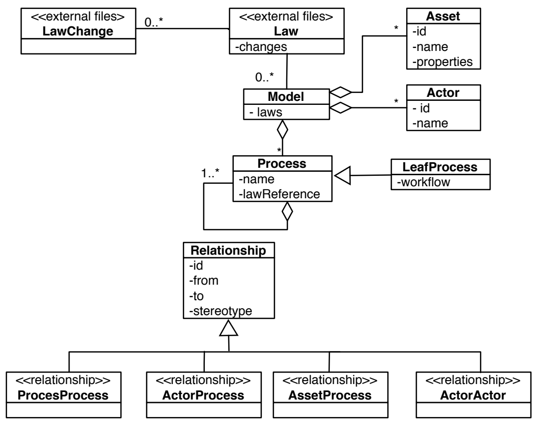

Fig. 1. Modeling Elements' Metamodel

A *process* is realized as an observable activity executed by one or more *actors*, which might be person, components, technical systems of combinations thereof. In particular, we use the definition of process as described in our UML modeling methodology [11] where processes are hierarchically organized. Each process is associated to law elements that define and regulate it. Leaf processes in the hierarchy contain the actual specification of processes as a sequence of activities —in our methodology they contain activity diagrams, written in a specific language that can be translated to UML or to asset flows on which formal analysis using the state of the art techniques can subsequently be performed.

An *actor* is responsible for a specific process or asset. Actors can be extracted from the text of the law or can be defined manually. Currently, our model identifies them by means of an unambiguous identifier (extracted from the XML file containing the law information) and a name. However, this could easily be extended in order to add more features, for instance, stereotypes.

Assets are what we focus on when performing formal analysis of processes and, in particular, security assessments. In the same way as actors, they can be either extracted from the law or defined manually. Assets are characterized by a feature which comprises of a set of properties, value and location. A particular instance of an asset feature defines its state. Asset flows describe the evolution of assets, i.e., possible sequences of states through which assets can proceed during its lifetime as a result of reacting to discrete activities performed on the asset feature (i.e., the value, location, and properties) and possibly other conditions meet. In our model, we store the initial states of assets and we use our notation to define the changes that the assets undergo.

Relationships are defined separately from the objects of the model. We use a generic relationship element to create specific sub-classes of relationships. *Actor-Actor* relationships have different properties from *Actor-Process* relationships (e.g., the allowed stereotypes) and from *Process-Process* relationships. To associate a process with its executing actors (i.e., *Actor-Process* relationship), we use the concept of *roles*. A role has a name $\in R$, where R is the set of role identifiers. 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 relationship to UML. We also explicitly support the *Asset-Process* relationships that define the semantics for the asset flows. In addition, the modeling borrows and extends some well-known formalisms from the RACIV responsibility matrix, and from the CRUD matrix, to support the different relationships discussed above.

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 that the law is not included in the model. For our case studies, we used examples from the Italian law system retrieved using the *Normeinrete* [15,16] project and its XML format. Although our metamodel is designed to support XML format for laws representation, it can be easily extended to support other formats.

4.2 System Components

We devised a 3-tier architecture based on a representation of the model which is independent from the tools used to visualize it and to perform formal analysis. In particular, the main system components are (see Figure 2):

1. **Data:** This contains the actual data on which the system operates as well as the visualization of models. In particular, it is responsible for comprising the process model (i.e., in XML model), XML representation of laws associated with processes in the model, UML modeling tool, and formal specifications' data (the *Intermediate Model*).
2. **Services:** Provide the core functionalities of the tool and contain the following components:
   – **Change Manager.** It provides all the functions to modify the model, import external modifications (e.g., amendments to the law regulating the processes being modeled) and analyze the impact of changes in the model to the law.
   – **Translation.** This is a class of components used to convert one form of model representation into other representations – namely, printable

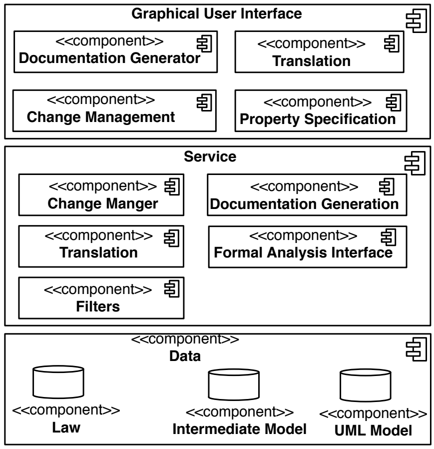

Fig. 2. The proposed VLPM$^+$ System's Reference Architecture

documentation and formal models – as well as extract procedures from the law and convert the changes in the model to possible changes to the law.

– **Documentation Generation.** It is responsible for generating human readable documentation of the model or of a subset of it.
– **Formal Analysis Interface.** It provides an interface to the formal analysis environment. This works together with a component that translates the workflow (in the form of activity diagrams) in the model to asset flows and subsequently to, e.g., NuSMV modules or other target formal tool. This part of the system also provides a service to assign feature values (e.g., asset properties) to the assets involved and to specify the security requirements that must be checked against the process model and asset flows. We do so by using the *property specification* GUI.

3. **Graphical User Interfaces.** They facilitate user interaction with the system. Since laws and processes are specified in natural language, some activities in the modeling workflow require some user intervention, e.g., in order to decide how to translate specific parts of a law in an ambiguous way, etc. More specifically, using respective interfaces the user is allowed to specify which subset should pass to the corresponding translator component (*Translation*), to specify critical requirements for formal analysis (*Property Specification*), to provide changes to the model based on the (formal) analysis report (*Change Management*), etc. Each of these components could pass through respective *filter component* – which allows the user to filter a complex model to produce a relevant information – before the subsequent steps.

5 Case Study

We used the Italian Immigration law as a case study for the preliminary analysis of the use cases of our integrated platform. Specifically, we analyzed the entrance

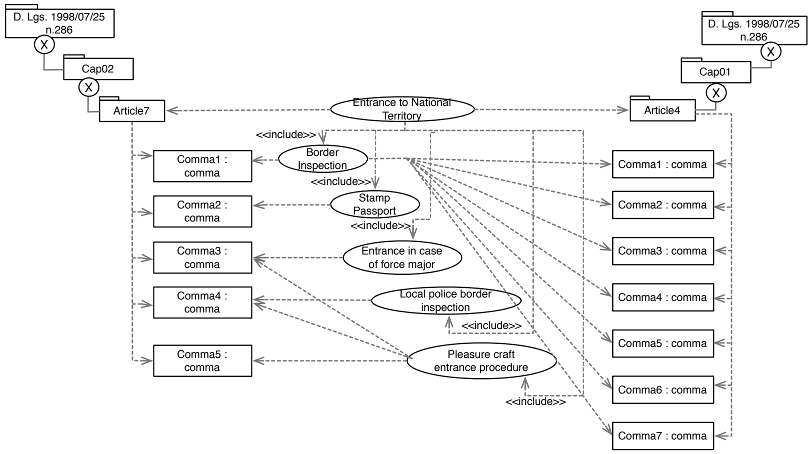

Fig. 3. Process Tree: Entrance into Italian National Territory

procedures at the Italian border and the required documents as defined in the 286^{th} legislative decree of July, 25 1998 on "Consolidated Act of Provisions concerning immigration and the condition of third country nationals" and in the 394^{th} presidential decree of August, 31 1999 (with all their changes until now). We considered the latter as the *primary law*, from which the process structure was extracted, and the former as an *additional law*, from which we derived extra dependencies.

We started applying our modeling methodology, which in this paper is simpler than the one presented in [11], in order to obtain the process decomposition tree shown in Figure 3. The figure highlights the *many-to-many* relationship among processes and the law paragraphs they are derived from or constrained by, but excludes actors which are involved in their realization. The two laws also provide information related to the order of the processes and related to the modification of assets (e.g., the *property* of the `Passport` was *"NotStamped"* and became *"Stamped & Inspected"*). See an example in Figure 4, which uses an activity diagram to refine the *"Border Inspection"* process.

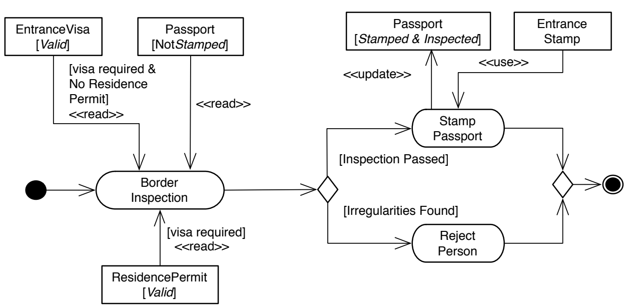

Fig. 4. Border Inspection.

Activity diagrams are strong as modeling means but lacks simulation and verification capabilities [9,10]. As a proof of concept, we manually formalized the diagram depicted in Figure 4 using Algebraic Petri Nets [17], an intermediate format composed of Petri Nets and abstract algebraic data types. Model checker like NuSMV [18] then can be used to perform formal verification and simulation.

The detail how we intend to perform the formalization, translation, verification and activities related to change management is not in the scope of this paper and thus we skip.

6 Related Work

In order to univocally identify laws and their elements, *NormeInRete* uses a URN (Uniform Resource Name) based system [15,19]. The URNs are defined as a combination of elements according to a specific grammar. The basic elements are: name of the promulgating authority, type of norm, date, number and, when needed, a set of more detailed specifications [15]. The authors in [20] presented xmlLeges application, an open source application suite for legal drafting.

The advantages and difficulties related to the (re-)engineering of public services are discussed in [1,2,3,6]. In [5,8,7], the authors particularly discussed the support of ICT for public *healthcare* services by identifying different levels of process support and distinguish between generic process patterns, as well as the level of automation. These approaches, however, lack the supporting tools and their corresponding precise methodology to perform the modeling (in some cases, to perform the analysis) of such processes. !

In [21] the authors propose a UML-based approach to define, verify, and validate organizational processes. More specifically, their modeling and verification are from the point of software process improvement through the CMMI framework (Capability Maturity Model Integration). The author in [22] discussed challenges and opportunities of using simulation techniques to reduce the risks and increase the chance for success of business process re-engineering. Clearly, these approaches are different from us in the sense that they did not particularly face the challenges of the public administration processes.

The importance of modeling in the legal framework and documenting the knowledge about the legal constraints within the process model itself is stated in [2,23]. 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). In [24] authors establish the basic need for formal techniques in electronic governance. The authors in [25] also discussed a pattern-based evaluation of UML 2.0 Activity diagrams for workflow modeling. These approaches can be refined and used in order to strength our approach.

7 Conclusion and Future Work

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 and their formal verification can take this even further. The modeling method itself must not be too complex since administrative executives are usually not familiar with modeling languages.

Although there are still some points need to further refine and develop, in this paper we discuss why it is difficult to adequately support PA processes by ICT systems and which challenges exist in this context. We proposed an integrated environment to overcome some of the mentioned challenges. We are currently implementing the various components discussed in this paper in order to integrate them with the current version of VLPM. In the future, we would like also to support the integration of PA processes on different levels through free simulation. We are also looking for case studies to evaluate our framework in a larger scope.

References

1. David, G., Wastell, P., White, P.: A Methodology for Business Process Redesign: Experiences and Issues. Journal of Strategic Information Systems 3, 23–40 (1994)
2. Alpar, P., Olbrich, S.: Legal Requirements and Modelling of Processes in e-Government. Electronic Journal of e-Government 3 (2005)
3. Xenakis, A., Macintosh, A.: Using Business Process Re-engineering (BPR) for the Effective Administration of Electronic Voting. The Electronic Journal of e-Government 3(2) (2005)
4. U.S.A. National Performance Review. Executive Summary — Creating a Government that Works Better and Costs Less (1993)
5. Willcocks, L., Currie, W., Jackson, S.: Radical Re-Engineering and Information Systems: Evidence from UK Public Services. In: Fifth European Conference in Information Systems, Cork (1997)
6. Thaens, M., Bekkers, V., van Duivenboden, H.: Business Process Redesign and Public Administration: a Perfect Match? In: Taylor, J.A., Snellen, I.T.M., Zuurmond, A. (eds.) Beyond BPR in Public Administration: An Institutional Transformation in an Information Age, pp. 15–36. IOS Press, Amsterdam (1997)
7. Greco, G., Guzzo, A., Pontieri, L.: Mining Hierarchies of Models: From Abstract Views to Concrete Specifications. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 32–47. Springer, Heidelberg (2005)
8. Lenz, R., Reichert, M.: IT Support for Healthcare Processes. In: van der Aalst, W.M.P., Benatallah, B., Casati, F., Curbera, F. (eds.) BPM 2005. LNCS, vol. 3649, pp. 354–363. Springer, Heidelberg (2005)
9. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Wohed, P.: On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling. In: Third Asia-Pacific Conference on Conceptual Modelling (APCCM 2006), vol. 53 (2006)
10. Eshuis, R.: Symbolic Model Checking of UML Activity Diagrams. ACM Trans. Softw. Eng. Methodol. 15(1), 1–38 (2006)
11. Ciaghi, A., Villafiorita, A., Mattioli, A.: VLPM: A Tool to Support BPR in Public Administration. In: ICDS, pp. 289–293. IEEE Computer Society, Los Alamitos (2009)
12. Object Managment Group. The Unified Modeling Language (UML2.0) Superstructure Specification. Technical report (2004)

13. Object Managment Group. Business Process Modeling Notation (BPMN). Technical report (2009)
14. Weldemariam, K., Villafiorita, A.: Formal Procedural Security Modeling and Analysis. In: CRiSIS 2008: Third International Conference on Risks and Security of Internet and Systems, Washington, DC, USA, pp. 249–254. IEEE, Los Alamitos (2008)
15. Lupo, C.: NormeinRete: a Federative Approach to On-line Legislation Access. In: International Conference Parliaments' Information Management in Africa: Challenges and Opportunities of ICTs to Strengthen Democracy and Parliamentary Governance, Nairobi, Kenya (2005)
16. Lupo, C.: Formato per la rappresentazione elettronica dei provvedimenti normativi tramite il linguaggio di marcatura XML. Allegato tecnico alla Circolare 22 aprile 2002, n. AIPA/CR/40 (2002)
17. Reisig, W.: Petri Nets and Algebraic Specifications. Theor. Comput. Sci. 80(1), 1–34 (1991)
18. Bertoli, P., Cimatti, A., Pistore, M., Roveri, M., Traverso, P.: NuSMV 2: An Open Source Tool for Symbolic Model Checking. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, p. 359. Springer, Heidelberg (2002)
19. de Oliveira Lima, J.A., Palmirani, M., Vitali, F.: 'HTTP" or 'urn' URIs for legal resources? How about both? In: Jurix 2007 Workshop on Legislative XML (2007)
20. Agnoloni, T., Francesconi, E., Spinosa, P.: XMLegesEditor: an OpenSource Visual XML Editor for supporting Legal National Standards. In: V Legislative XML Workshop, pp. 239–251 (2007)
21. Hsueha, N.-L., Shen, W.-H., Yanga, Z.-W., Yanga, D.-L.: Applying UML and software simulation for process definition, verification, and validation. Information and Software Technology 50, 897–911 (2008)
22. Hlupic, V.: Business Process Modelling Using Discrete Event Simulation: Potential Benefits And Obstacles For Wider Use. International Journal of Simulation: Systems, Science and Technology 4, 62–67 (2003)
23. Olbrich, S., Simon, C.: Process Modelling towards e-Government – Visualisation and Semantic Modelling of Legal Regulations as Executable Process Sets. Electronic Journal of e-Government 6 (2008)
24. Davies, J., Janowski, T., Ojo, A.K., Shukla, A.: Technological foundations of electronic governance. In: ICEGOV, pp. 5–11 (2007)
25. Russell, N., van der Aalst, W.M.P., ter Hofstede, A.H.M., Wohed, P.: On the Suitability of UML 2.0 Activity Diagrams for Business Process Modelling. In: Third Asia-Pacific Conference on Conceptual Modelling (APCCM 2006), vol. 53 (2006)