Modeling E-Government Processes Using YAWL

Half-way Towards Their Effective Real Implementation

Orlando Belo, José Luis Faria, António Nestor Ribeiro
Department of Informatics
School of Engineering
University of Minho
PORTUGAL

Bruno Oliveira, Vasco Santos
CIICESI
School of Management and Technology
Polytechnic of Porto
PORTUGAL

ABSTRACT
Today E-Government institutions face a lot of challenges related to the quality and effectiveness of the services they provide. In most cases, their users are more demanding, imposing new ways of acting and dealing with their needs, requesting often expeditious and effective attendance. Independently for their nature, we believe that such pertinent characteristics begin to be sustained immediately as we start to study and model E-Government processes. Modeling and simulation are useful tools on the assurance of the availability of E-Government services in many aspects, contributing significantly to improve processes implementation, ranging from their inception to their final software application roll-up and maintenance. In this paper we studied the use of YAWL – a work flowing language – for modeling E-Government processes, showing through a real world application case how it can help us in the construction of effective models that may be used as a basis for understanding and building the correspondent software applications.

Categories and Subject Descriptors
I. Computing Methodologies; I.6.5 Model Development - Modeling methodologies.

General Terms
Documentation, Design, Standardization, Languages.

Keywords
E-Government process modeling, conceptual modeling, workflows representations, YAWL.

1. INTRODUCTION
Nowadays, the virtualization of governmental services is not anymore as novelty, it is a necessity. E-Government applications came to define a new kind of relationship between public institutions and the citizens, independently where they are located.

The goal is quite clear. The implementation of E-Government services approach citizens to their public responsibilities as well as facilitates dialogue between them and public institutions through ubiquitous services. Many governments have been promoting and implementing the use of E-Government applications, in order to ensure the already mentioned citizens “proximity” and improve their quality of service and effectiveness [3]. As any other case of using digital technologies, E-Government services implementation requires effective practices of software design and implementation, in order to map accordingly processes and tasks that one intend to be used directly by citizens or, in a more restrictive case, by the employees of a public institution, like it usually happens in *Government-to-Government* (G2G) scenarios. However, we know that any piece of software to be effectively used must be designed accordingly, following the best practices of software development [12]. That includes using modeling techniques in the first stages of development trying to capture all the requisites (functional and operational) that will sustain a specific process and regulate interaction (and communication) of its users. Modeling is a key task in the process of gathering the quality and effectiveness of service E-Government requires.

Today’s modeling tools market offers a large diversity of modeling languages, presenting features covering the needs of the most critical applications in retail, telecommunications, banking, and, of course, E-Government. Thus, selecting one of them is not an easy task. However, if we focus on the description of the dynamic of a process (or a task), and how interaction happens among different processes, we choose a work flowing language naturally. Among them, and taking our experience in other application fields [10], we choose the YAWL language [1], mainly due to its simplicity of usage and formalism for workflow representation and orchestration. We want to demonstrate that we can improve quality and effectiveness of E-Government services through a better conceptualization of their processes, and consequently their implementation, independently that we are starting a new E-Government information system project or refactoring a pre-existing one. To prove the feasibility of our approach, we used the YAWL language for the specification of the data related to the execution of a model for a particular application case of an external service authorization process of a public university. In practical terms, we may classify this case as a typical G2G service. Thus, in this paper we present a brief exposure of some pertinent issues about conceptual modeling techniques (section 2), in general, and their application to E-Government processes and services. In particular, we present and discuss a process model in YAWL that we designed and developed for a specific application case we deal frequently in a department of a public university (section 3), discussing some of their internal tasks, depicting them with some relevant workflow processes. Finally, we present some conclusions discussing the most pertinent aspects presented, and pointing out some ideas to be explored in a near future.

2. Process Modeling

The choreography requirements that E-Government process models must comply according the demands of external business partners, represent a critical step in any transactional service implementation. Several works address this issue providing methodologies that support initial design steps and guide users to more specific technical requirements. Several proposals using UML (Unified Model Language) were presented to represent organizational processes. For example, Kim [6] used UML activity diagrams, demonstrating how these conceptual models can be transformed in ebXML specifications. Still using UML, Tyndale-Biscoe et al. [14] provided an UML profiler that allows for the definition of specific mappings between business concepts and software artifacts. These authors argued that business models must be an integral part of a system model, providing translations between more abstract and concrete technical specificities. Kramler [8] proposed an UML modeling technique that is based in three levels of abstraction: collaboration level, transaction level and interaction level, each one of them defining a different abstraction level in process orchestration modeling. A set of mappings to transform modeling concepts to BPEL is also addressed in that work. In turn, Palkovits [11] proposed a holistic approach to support process analysis, re-organization and modeling. Also considering some of these specific requirements, the UN/CEFACT's Modeling Methodology (UMM) [9] was proposed as the top down approach to capture business requirements, providing a formal approach to the definition of both organizational and technical aspects. This methodology was provided as a specific subset of UML to capture business collaborations. The authors also addressed the representation of more general UMM models to the executable primitives, providing its translation to BPEL (Business Process Execution Language) primitives. Due to many reasons, such as political and rigid structures, changing requirements is a very common task in the public sector. Stemberger et al. [13] presented several important aspects associated to such situation inside public sector institutions, their specificities, and how the methodology they proposed can be applied in order to support process changes. Chebbi et al. [4] proposed a view-based approach to dynamic inter-organizational workflows. Based on the BPMapping [2] and UMM, Chourabi [5] proposed an expressive modeling approach for business processes in government applications. The BPMN (Business Process Model Notation) was also already explored for the execution of inter-organizational processes [7]. In the referred work, authors expressed business requirements using BPMN, through the use of collaborative processes, to represent conceptual models and process execution.

With this work, we focus on the technical aspects of how to model effectively organizational processes, proposing to do that using of the YAWL workflow language [1]. YAWL provides a formal and intuitive way to represent workflows. We cover two important aspects associated to this type of business processes: conceptual representation with an understandable notation, which contributes to improve communication between stakeholders, and process execution and validation, separating concerns between operations orchestration and data involved. YAWL also has the ability to receive additional features, such as multiple instance support and cancellation patterns, which contributes to the creation of more detailed models for complex workflows. All these formalisms are very useful since they make the language more concrete contributing to less ambiguity. Additionally, YAWL supports exception handling, dynamic workflows, declarative workflows and a powerful and simple notation to represent all of its constructs. With a particular case study, we explore YAWL application and suitability when applied to the needs and specificities associated to E-Government processes.

3. Modeling E-Government Processes

3.1 The Application Case

In order to demonstrate our YAWL approach to model E-Government processes, we selected an *external service authorization* (ESA) (or mission, for short) process of a public university - a typical G2G service. We will characterize and discuss it based on the experience we acquired when we need to model and implement it accordingly the specificities of department of that institution. In Portugal, any public institution must follow strict administrative procedures regarding all public processes. The ESA is one of those procedures, which was defined to regulate the way how the members of a department participate in scientific and technical events, such as conferences, meetings, workshops, and so on, outside Portugal. To be authorized, every exit abroad demands the definition a specific package of data, quite complete, involving a lot of attributes, namely: date and hour of leaving and return, destination, the description of the event or service, a detailed budget, the list of the institutional services affected, just to name a few. Until 2008 this process was made entirely in paper, having so a lot of well-known restrictions – e.g. error prone, uncontrolled redundancy, integration checking difficulties, or difficult data exploitation and communication. However, since 2009 this process runs digitalized and follow a specific workflow, especially designed with the purpose to facilitate communications between all the entities involved with, within the department and in the university. An ESA process starts with a very conventional processing task. The person who wants to apply to participate in a conference, for instance, needs to fill a pre-established form. This form is provided by a web application that beyond this initial task supports as well all the subsequent steps in the process. Obviously, as expected, this "new" way of doing things revealed some real advantages: the fill of the form is not error prone anymore, simply because the operational system is now responsible to do the most critical tasks – integrity constraints assurance, calculation, inter process communication, etc. Additionally, others benefits emerged naturally like as reverberating the budget on expenses accounts, giving instantly access to available balances, providing an availability of service of 24 hours a day, 7 days a week, or disposing easily search and find data services across operational databases. Usually, an ESA process involves four entities: 1) an applicant who fill the form; 2) a person of the secretariat that validates the form and perform other administrative procedures; 3) a manager account (it could be involved more than one) that checks the budget and authorize (or deny) the expense, and, finally, 4) the head of department who settle all the process. All these entities have their roles perfectly defined as well as the moments when they must act on the process. In Figure 1, we can see a general description in pseudo code of the process making reference of its most relevant parts.

```
process ExternalServiceAutorization (person,result)
begin
  fulfillApplicationForm(person,form)
  secretariatValidation(form,validation)
  if validation=true
   then managerAccountValidation(form,validation)
    if validation = true
     then headOfDepartmentValidation(form,validation)
      if validation = true
        then authorizeExternalService(person,form)
             informApplicant(person,form,validation)
      endif
    endif
   else informApplicant(person,form,validation)
  endif
end
```

Figure 1. ESA process pseudo-code description

3.2 Modeling in YAWL

To model the ESA process, we need to consider and reflect the representation of specific aspects related to some organizational requirements. Using YAWL we have the ability to provide very detailed models describing the interaction that may exist between several entities that are associated to specific roles with specific permissions. YAWL provides a service-oriented architecture, giving us a great flexibility in process specification, keeping the separation of concerns between workflow coordination and task processing. For example, it is possible to associate specific tasks to web services calls, enabling the use of particular features such as, email or SMS notifications. Additionally, tasks can easily be assigned to human participants, Java code and external applications. YAWL provides powerful data perspectives with the use of standard and well-proven technologies for data manipulation: XML for the data representation, XPath (XML Path Language) and XQuery (An XML Query Language) for data extraction and manipulation. Furthermore, it also provides a very useful service called Selection service that has part of the Worklet Service. With this service we can replace a work item in a YAWL process specification at runtime. YAWL models are organized as nets that represent specific process parts at different abstraction levels. Basically, we can represent our general net for an ESA process with three composite tasks (Figure 2): 1) “Mission Proposal”, which represents the submission and validation process over the request done by an applicant); 2) “Processing Expenses” representing tasks for the processing of expenses); and 3) “Reporting”, which represents all tasks that should be completed in order to submit the final mission report.

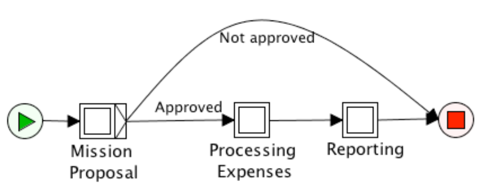

Figure 2. ESA Process View – Level 1

The “Mission Proposal” task has two possible states: ‘Approved’, which allows the workflow to proceed with the execution of successful path, and ‘Not approved’, which results in the end of the process. This process represents a very abstract view of the whole process (Level 1). If the “Mission Proposal” task is zoomed in, we access to a more detailed view of the process (Figure 3).

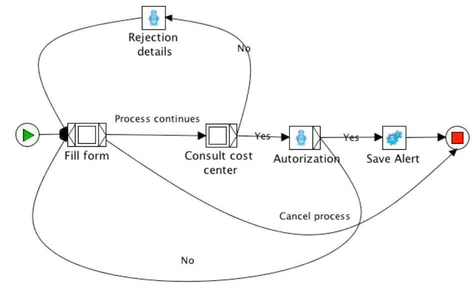

Figure 3. ESA Process View – “Mission Proposal” - Level 2

The process presented in Figure 3 is a more detailed process. It is a Level 2 process. It was built using a set of atomic tasks: “Rejection details”, “Authorization” and “Save Alert”; and two other composite tasks: “Fill form” and “Consult cost center”. The process starts invocating a composite task that is responsible to receive the data that should be entered by the applicant. Next, it is performed the evaluation of the mission (the external service). This evaluation task (composite) represents a set of more detailed procedures that must be performed to achieve a specific “answer”, which is the communication of a decision. Based on this answer two possible states can be identified: the incorrect mission submission, which implies the generation of specific corrections that the original applicant must do, or it can imply the invocation of authorization task, which involves, in turn, the head of responsible. These tasks can also split the main flow in two possible directions: ‘Acceptance’ and ‘Rejection’. The top-level manager can reject the mission by various reasons, indicating that the original data must be corrected and re-processed again by all previous tasks. Note that “Fill form” composite task can cancel the entire process, due to the rejections reasons provided by “Consult cost center” or “Authorization task”. If the authorization is granted, a specific automated procedure is launched in order to inform the applicant that its mission was accepted. In Figure 3, we can see a process that provides a more detailed view of the tasks that are included in the ‘Mission Proposal’ task (Figure 2). However, we can continue to “drill-down” our model in order to access to a more detail representation of the process tasks. Figure 4 presents an internal specification of the ‘Fill form’ composite task (Level 3). The process depicted in Figure 4 describes the main activities presented in the mission description. In this model, the applicant fills general data and sets of verifications that must be made in order to identify optional data to be included in the mission description

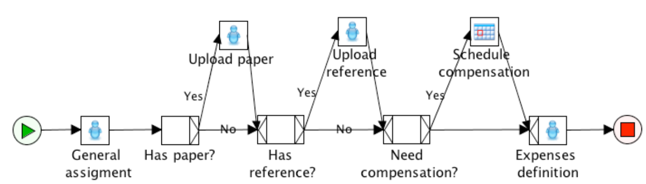

Figure 4. ESA Process View – “Fill Form” – Level 3

Most of these activities are completed by a participant and can include additional validations in its internal specification. The model presented in Figure 3 also represents the ‘Consult Cost Centre’ task, which represents the main activities related to the validation of each expense identified in previous processes (Figure 5).

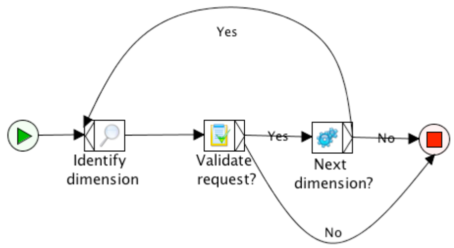

Figure 5. ESA Process View – "Expenses Validation " – Level 3

In Figure 5, we can see that the process starts with the dimension costs processing for all expenses. Next, the secretariat validates the request in conjunction with each of the dimensions' responsible, checking if it is possible to allow the expense. This process is repeated for every dimension. The remaining two composite tasks referred in Figure 2 ("Processing Expenses" and "Reporting tasks") can also be decomposed in the same manner as we did for the "Mission Proposal" task. With this kind of specification we can represent clearly the main workflows, providing possible paths that specific resources we can execute. Thus, we need to specify the resources and data requirements for each task. Firstly, we describe the main roles in the process: applicants, managers, secretariat members and head of department members. For each of them we can instantiate members using the configuration defined for each role. Furthermore, it is possible to allocate resources for specific tasks based on interaction strategies that define authorization policies for the interaction of participants. Secondly, YAWL processes are capable of capturing all necessary metadata for process execution and can not only provides all tasks orchestration but also represents and control all necessary metadata involved between all tasks. Thus, it will be described how participants will interact with tasks, concerning the input and output data needed for each task. The interaction with other tasks from the same net is configurable through the decomposition of tasks and the definition of specific input and output parameters, as well as their data types.

4. CONCLUSIONS AND FUTURE WORK

YAWL provides us a very clear way to represent complex business processes in different layers, which can be applied successfully to E-Government processes using a top-down approach. This is very useful in information system conceptualization, since it abstracts in a very simple manner very complex processes. It's a clear advantage for high level users when presenting, discussing and understanding systems models that represent real world application processes. They don't need to have an overall view of the remaining tasks. YAWL provides a simple and very broad notation that coupled with powerful execution primitives and data support structures turns the representation of E-Government processes very suitable, not only for conceptual representation but also to process execution. Many other YAWL features were not covered in this paper, like the C-YAWL. These YAWL specific models provide a way to specify configurable tasks in order to establish variants in some parts of the process, which makes it adaptable for different application scenarios. As soon as possible, we will extend our approach to other operational units, working on another set E-Government processes, defining their conceptual models in YAWL, formalizing them accordingly to current quality enforcement procedures, and when possible transform the models created in applications skeletons using execution primitives, taking into consideration the possibility to incorporated them in one of the operational information systems. To do that, "model-to-code" transformations will be designed and applied to such cases to allow their representation in some machine understandable code that can be posteriorly translated and execute.

5. REFERENCES

[1] van der Aalst, W., Ter Hofstede, A., "YAWL: Yet Another Workflow Language," Inf. Syst., vol. 30, pp. 245–275, 2003.

[2] Brandenburg, H., Wojtyna, J.-P. , "L'approche processus: mode d'emploi". 2003, p. 192.

[3] Brifing Note 3 May, "e-Government Applications", United Nations Asian and Pacific Training Centre for Information Technology and Development (APCICT), May, 2010.

[4] Chebbi, I., Dustdar, S., Tata, S.,"The view-based approach to dynamic inter-organizational workflow cooperation," Data Knowl. Eng., vol. 56, pp. 139–173, 2006.

[5] Chourabi, H., Mellouli, S., Bouslama, F., "Modeling e-government business processes: New approaches to transparent and efficient performance," J. Inf. Polity - Gov. Inf. Shar. Integr. Comb. Soc. Tech. Pap. from 9th Int. Conf. Digit. Gov. Res., vol. 14, no. 1,2, pp. 91–109, 2009.

[6] Kim, H., "Conceptual Modeling and Specification Generation for B2B Business Processes Based on ebXML," in SIGMOD Rec, 2002, pp. 37–42.

[7] Kim, H., "BPMN-Based Modeling of B2B Business Processes from the Neutral Perspective of UMM/BPSS," in IEEE International Conference on e-Business Engineering, 2008.

[8] G. Kramler, G., Kapsammer, E., Retschitzegger, W., Kappel, G., "Towards Using UML 2 for Modelling Web Service Collaboration Protocols," Interoperability Enterp. Softw. Appl., pp. 227–238, 2006.

[9] Liegl, P., Schuster, R., Zapletal, M., Huemer, C., Hofreiter, B., Mosser, R., "Modeling eGovernment Processes with UMM," Informatica, vol. 31, pp. 407–417, 2007.

[10] Oliveira, B., Belo, O., "ETL Patterns on YAWL - Towards to the Specification of Platform-independent Data Warehousing Populating Processes", In Proceedings of 16th International Conference on Enterprise Information Systems (ICEIS'2014), Lisbon, Portugal, 27-30 April, 2014.

[11] Palkovits, S., Orensanz, D., Karagiannis, D., "Process Modelling in E-Government Living Process Modelling Within a Public Organisation" in IADIS International Conference e-Society, 2005.

[12] Sommerville, I., "Software Engineering", Addison-Wesley, 9th edition, March, 2010.

[13] Stemberger,M.I., Jaklic, J., "Towards E-government by Business Process Change - A Methodology for Public Sector," Int. J. Inf. Manage., vol. 27, pp. 221–232, 2007.

[14] Tyndale-Biscoe, S., Sims, O., Wood, B., Sluman, C., Business Modelling for Component Systems with UML", In Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC'02. IEEE Press 2002, 2002.