Livelock and Deadlock Detection for PA Inter-organizational Business Processes

Damiano Falcioni, Andrea Polini, Alberto Polzonetti, and Barbara Re

Computer Science Division
School of Science and Technologies
University of Camerino
62032 – Camerino (MC), Italy
firstname.lastname@unicam.it

Abstract. The Public Administration domain is characterized by the dominance of inter-organizational Business Processes. These are a set of interrelated and sequential activities shared and executed by two or more Public Administration offices to achieve a business objective that is of value to citizens or companies in term of services. A Business Process results from the un-trivial integration of internal administration processes, so that structural problems such as livelock or deadlock may easily occur and in reality they are generally solved by involved civil servants. Nevertheless with the shift versus an electronic government this problem becomes particularly relevant. The paper presents a suitable approach for inter-organizational Business Process detection of livelock and deadlock situations. In particular, we introduce an approach to directly verify a Business Process modeled using the BPMN 2.0 semi-formal notation. The verification uses a state evaluation technique with an optimized unfolding algorithm considering specific BPMN 2.0 characteristics. A plug-in for the Eclipse platform has been also developed, which permits to have an integrated environment in which to design Business Process, using the Eclipse BPMN 2.0 Modeler, and to automatically verify it. The approach and the tool prototype have been successfully applied to real scenarios such as family reunion, grant citizenship and buoncer registration.

1 Introduction

Due to the maturity of the European Interoperability Framework [1] and its adoptions in European member states we realize that one of the main challenge in Public Administration (PA) is to cope with large collections of interconnected Business Processes (BPs). Technical and organizational interoperability should be embedded in the design of e-government information system. This is a problem both at national level, where different Public Administrations have to cooperate in order to provide a service, and at European level, where the development and improvement of cross-border e-government services have to become a reality according to the EU Ministerial Declaration on e-government [2] as well as to the

A. Kö et al. (Eds.): EGOVIS/EDEM 2012, LNCS 7452, pp. 125–139, 2012.

European Digital Agenda that, among the different objectives, aims at creating one single market in Europe [3].

According to the proposed motivational scenario, and mainly focusing on organization interoperability, we believe that Business Processes modeling and analysis gain more and more importance facing with the need of efficient and effective definition of inter-organizationals Business Processes. Conventional Business Process Management (BPM) research mainly dealt with intra-organization processes that in most of the cases are isolated. Even if value chains and boundary phenomenon have already been studied, collaboration in the context of BPM is still a topic under growing research and so far little research has examined the implication of boundary blurring BPs [4].

Now the challenge is to move toward an open environment [5] where large organizations, such as the Public Administration, have hundreds of BPs in place. Inter-organizationals BPs result from the un-trivial integration of internal organization processes, so that structural problems such as deadlock and livelock may occur. We believe that formal methods and in particular verification can find a interesting application field, in order to make PA more effective and efficient.

In this paper we present an approach for formal verification of PA inter-organizational BPs. The approach uses state evaluation techniques with an optimized unfolding algorithm based on BPMN 2.0 specific semantic. In this way after BP modelling using BPMN 2.0 the analyst can run our algorithm to check if the BP includes bad traces. In such a case he/she can re-engineer the BP in order to remove the bad traces so to have an improved BP.

A plug-in for the Eclipse platform has been developed. It permits to have an integrated environment in which to design Business Processes, using Eclipse BPMN 2.0 Modeler, and to automatically verify the process model via the proposed algorithm. The approach and the prototype have been successfully applied to real scenarios thanks to a close collaboration with local PA offices, with encouraging results. In particular, we report here the three cases studies: (i) family reunion, (ii) grant citizenship, and (iii) bouncer registration, which have been the subjects of our investigations.

The rest of the paper is organized as follows. The next Section presents background information, whereas Section 3 introduces the verification approach we propose. Section 4 presents the case studies and Section 5 describes the results obtained from the conducted experiments. Concluding remarks and opportunities for future developments are discussed at the end of the paper.

2 Background

2.1 Process Modelling

Technically public service related processes can be modelled and implemented using notations and tools based on the BP concept.

> "*A BP is a collection of related and structured activities undertaken by one or more organizations in order to pursue some particular goal.*

Within an organization a BP results in the provisioning of services or in the production of goods for internal or external stakeholders. Moreover BPs are often interrelated since the execution of a BP often results in the activation of related BPs within the same or other organizations". [6]

In addition to the BP concept collaborative BP represents an issue in order to reach the suitable point of view able to represent the right abstraction level [7]. Recent works show that BP modelling has been identified as a fundamental phase in BPM. The quality of BPs resulting from the BP modelling phase is critical for the success of an organization. Its importance exponentially grows in order to support inter-organizational processes and related service delivery. Different classes of languages to express BPs have been investigated and defined. There are general purpose and standardized languages, such as the BPMN 2.0 [8] or the Event-Driven Process Chain [9] and many others. There are also more academic related languages, being the Yet Another Work-flow Language [10] the most prominent example.

In our work we refer to BPMN 2.0 [8] an Object Management Group (OMG) standard. This is certainly the most used language in practical contexts also given its intuitive graphical notation. We mainly use collaboration and conversation diagrams in order to have a complete representation both of internal process as well as of the message exchange.

2.2 Formal Verification

In the context of software systems, formal verification is the act of proving or disproving the correctness of a system with respect to a given formal specification or properties, using methods based on sound mathematical tools. Many different formal approaches can be applied to systems verification. Some studies have been reported on the application of formal methods in e-government [11] in order to analyse Business Processes. Standard approaches mainly refer to two main categories referring model checker for petri nets and process algebra, so there is the case that dedicate mapping has to be implemented in order to make formal verification accessible to the wide audience and integrated in the whole BPM life-cycle. The mapping mainly results with constraints on the process model and expressibility that can be represented during the modelling phase can be reduced due to the semantic of the target language. For instance, using BPMN 2.0 as modelling language if we consider petri-net as target language could happen that information about pools are lost. On the other side if we use process algebra as target language syncronization is a must also in the case it is not strictly needed, in fact using sequential step instead of synchronization inside a process we cannot map cycles. Common problem is state explosion [12].

In our work we propose a BP verification technique based on an optimized unfolding algorithm taking advantage of the specific BPMN2.0 semantic. In such a way we avoid problems resulting from the application of a mapping to a formal language. In order to avoid state explosion phenomenon the proposed optimization bound the branching with reference to exclusive gateway.

2.3 Unfolding

Unfolding is a technique of partial order reduction. It is widely applied to Petri Nets and Process Algebra in order to reduce state explosion problems during verification. Unfolding has proved to be very performant mainly on deadlock detection. It is based on the concept that some decidability problems can be reduced to reachability problems (proved to be decidable in many research works like [13]). In order to solve the problem, a prefix of the model is built with the objective to cover all the reachable states.

Unfolding of a model can in fact be infinite, but McMillan [14] identified the possibility of building a finite prefix of the net which could give us enough information to solve several problems. This is made ending the prefix in a specific point called "*cut-off*".

The concept of configuration is introduced, it is used to identify the current status of the model referred to a specific path, during the unfolding. So the key to terminate the unfolding is to identify configurations states acting as cut-off points. This must have the following property: any configuration containing a cut-off point must be equivalent (in terms of final state) to some configurations containing no cut-off points. From this definition, it follows that any successor of a cut-off point can be safely omitted in the unfolded model, without sacrificing any reachable state of the original model. This means that a cut-off point is reached in a configuration only when the current final state of the configuration is already present in the same configuration as previous state. So this current final state is our cut-off point. If we do not stop the prefix of the unfolded model in this cut-off point, we are sure that this state will infinitely happen again in the future because it has already happened once in the past.

Once the prefix has been constructed, the deadlock detection problem is reduced to a graph problem. This problem is NP-complete as shown by [14]. However this problem is readily solved in practice even for very large unfoldings. Problems, such as liveness and deadlock-freeness, in fact, are recursively equivalent to reachability, so that they are also decidable [15]. Approaches are also proposed in literature to verify temporal logics through unfolding [16].

There are several facts that are worth mentioning about the unfolding. The first is that the unfolding is an acyclic graph, defining a partial order on its nodes. Second, branching occur naturally in the structure where actual choice occurs (total order perform unneeded choice on unrelated transitions). The advantage is that we can explore the state space of concurrent systems without considering all possible interleaving of concurrent events. This became particularly significant if in the model there is a cycle.

In this work we base our verification approach on an optimized unfolding algorithm that exploits advantages provided by the use of BPMN 2.0. In particular, we reduce the state explosion problem reducing the interleaving between BPMN 2.0 elements, in a way that:

– Synchronizations happen only on parallel gateway in a pool and messages exchange between pools;
– Branching in the configuration tree happens only with reference to exclusive gateway.

The following section provides further details on the approach.

3 Proposed Approach

The approach we have defined supports Java oriented verification on BPMN 2.0 collaboration models. Starting from the results of a BP design phase we explore the model and we identify paths from BPMN 2.0 start events reaching a termination condition. Path definitions rely on synchronization and branching rules that are affected by the semi-formal semantic of the following BPMN 2.0 elements: pool, sending and receiving tasks, and parallel and exclusive gateways.

As already mentioned, in the collaboration diagram each pool encapsulates all the private process elements and a pool can interact via message exchanges with other participants. The approach we propose uses such encapsulation in order to eliminate interleaving among the elements in different pools that do not require synchronization. Synchronization is needed just for those tasks and events receiving or sending a message. Moreover, synchronization is observed in a pool for parallel gateways with references to input flows. Exploring the BP model, branching occurs only with reference to exclusive gateway. This give us the possibility to explore alternative paths. The specific characteristics of BPMN 2.0 elements suggests to conceive an optimization of the McMillan unfolding algorithm [14] specifically adapted to work on BPMN 2.0. In order to do that, we reviewed keys concepts such as configuration and cut-off points in a BPMN 2.0 context.

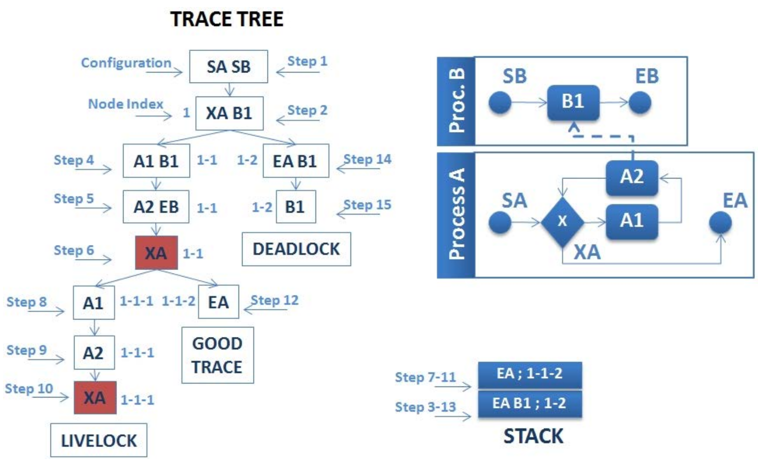

Fig. 1. Running Example

In Figure 1 we report a simple BPMN 2.0 collaboration model in order to make clear to the reader relevant aspects of the approach we propose. The BP is composed by two participants (pools) that exchange a message, the participant A internally decides how to behave according to the evaluation of the choice statement.

3.1 Configuration Definition

A configuration represents a possible partial run of the BP model under study. It is used to identify its current status during the execution of the unfolding algorithm. In Petri Nets the configuration contains the marked transitions. With reference to our approach a configuration refers to a specific path in the BP model. It differs from Petri Nets because in our case configuration contains all the activated BPMN 2.0 elements included in the collaboration diagram and their status.

In our approach all the configurations are stored in a tree structure. This is inspired from the coverability tree for Petri Nets introduced by Karp and Miller [17]. The coverability tree is an abstraction of the reachability tree which is precise enough to decide some important problems like coverability, boundedness and place boundedness problems for Petri Nets, and that were shown to be decidable [17]. Each node of the tree contains the BPMN 2.0 elements in the collaboration diagram that are currently active in term of diagram exploration.

Moreover, the application of unfolding algorithm in BPMN 2.0 satisfies the following conditions:

- If an element is in the configuration, then all of its ancestors are in the configuration too (a configuration is downward closed);
- A configuration can not contains two BPMN 2.0 elements in conflict, meaning that both are inputs from the same exclusive gateway.

The status of the BPMN 2.0 element is mainly relevant for what concerns those tasks sending and receiving messages. In particular, we accept as valid only those messages coming from tasks sending or receiving messages from an ancestor node in the configuration. We choose to evaluate such status at run-time instead of saving it in the configuration tree, in order to reduce memory consumption. As an example we refer to the steps 9 and 15 of the BPMN 2.0 exploration in Figure 1. During step 9 a message is sent from A2 task to B1 task, but it is never consumed because the task B1 has been already executed. As an effect during the backtracking (in step 13) B1 has a message that could be consumed but such messages do not come from an ancestor node of the current execution status so that such message is invalid and the B1 task remains blocked.

In order to correctly do the backtracking activities and complete the exploration of the tree we use a stack. It contains the tree nodes that have to be reloaded during backtracking and the index of the tree structure referring to the position in the tree of such nodes. The index is in the form *1-n-n-...-f*, parsed from left to right, where:

- *1* represents the root node and all the following nodes that do not split the tree;
- - separates between splitting nodes;
- n is a number representing the following node chosen from the current one and it includes all the following nodes after that choice till the next one;
- f represents the last branching node index and it is the same for all the following node until the path is ended.

This index structure makes easy the run-time evaluation regarding messages synchronization is particularly useful when there are messages sent from a task in an ancestor node of the configuration.

3.2 Livelock Identification (Cutoff Points)

The approach we propose exploits the configuration tree in order to find out cut-off points to identify livelock situations. A path is in livelock iff the current node is already observed during the exploration phase in one ancestor node of the configuration tree. In the case there are not cycles in the BP model then it is obvious that cutoff points can not be observed. In the configuration tree in Figure 1 the node referring step 10 is an example of cutoff point. As matter of fact the node involves the element XA and it is already observed in the ancestor node with reference to step 6.

3.3 Deadlock Identification

The approach follows the BPMN 2.0 termination paradigm in order to find out a deadlock. In BPMN 2.0 a BP terminates when end or termination events are reached during the process. The approach we propose adapts the unfolding algorithm in order to remove end events from the configuration each time they occur. The path results in deadklock iff in the current configuration there are only blocked elements (i.e. task or events waiting a message and parallel gateway waiting incoming flow that will never arrive). In the example in Figure 1 we observe a deadlock in step 13 when after backtracking, due to the gateway XA, the end event EA is reached and it is removed and the task B1 is blocked waiting for a message that will never arrive.

3.4 Good Trace Identification

The application of the approach returns with a good path when removing all the end events from the configuration then the tree node results to be empty. A node can be also emptied if a termination end event occurs. This means that the path is good and the process execution will stop correctly.

3.5 Prototype Implementation

The approach we propose is supported by a plug-in for the Eclipse platform permitting to have an integrated environment in which to design and verify BP models. The plug-in architecture is shown in Figure 2.

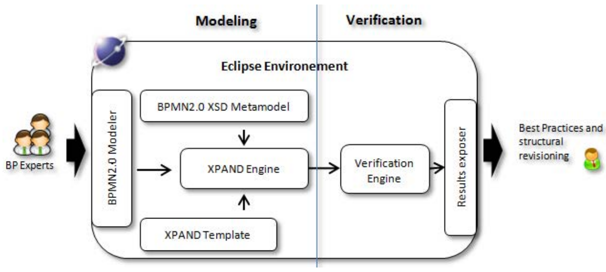

Fig. 2. Tool Architecture

The plug-in integrates the Eclipse BPMN 2.0 SOA Modeller[1] and is built over the BPMN 2.0 meta-model defined using the Eclipse Modeling Framework (EMF)[2], which has been defined respecting the official XSD schema for the standard.

The plug-in also integrates the XPAND[3] Eclipse engine. This is a language specialized for code generation which is based on EMF models and on the definition of transformation templates. This means that we are able to directly generate Java artefacts starting from a BPMN 2.0 model. The templates specifies what to do when a given graphical element of the BPMN 2.0 notation is found in defined BP model. Thanks to the XPAND engine the Java model of the Business Process is returned in output, so that it is possible to successively manage the model as Java objects. Verification runs on such a model implementing the algorithm. This is supported by the verification engine that gives back information about deadlocked, livelocked and good traces in textual way. Information about number of nodes checked, memory and time used are also shown. The engine has been built trying to minimize memory consumption and algorithm efficiency so that the Java Virtual Machine (JVM) interpretation process maximize performances.

The developed prototype can be downloaded from the repository hosted by Sourceforge[4].

4 Case Studies

The work we present refers to three real case studies concerning PA provided services. All of them are examples of inter-organizational BPs with several interactions between PAs. In particular, the considered services are:

– ***Family reunion*** – this is a service available for people legally residing in Italy which can apply on behalf of their relatives (spouse, depending parents,

[1] `http://www.eclipse.org/bpmn2-modeler/`
[2] `http://wiki.eclipse.org/MDT-BPMN2`
[3] `http://wiki.eclipse.org/Xpand`
[4] `https://sourceforge.net/projects/cowslip`

children less than 18 years old) for the purpose of family reunion and only after having provided evidence of their status with respect to “sufficient” incomes and a permanent address.

– ***Grant citizenship*** – this is a service used to ask for Italian citizenship by a foreigner or stateless person who has married to an Italian citizen or who is continuously residing in Italy for more then ten years.

– ***Bouncer registration*** – this is a service used to register bouncers in order to carry on their activity within public places.

The first and the second services require complex and inter-organizationals BPs and they are in place for several years now, therefore can be considered deeply tested. To give a quantitative indication in 2010 the Prefecture of Ancona (the capital city of Marche Region, in Italy) received 469 applications for family reunion and 760 applications for granting citizenship. For what concerns the bouncer registration service, even if it presents a simpler scenario, we choose it because its deployment is still on-going. We had the opportunity then to intervene and contribute to its development. In the following we provide a general description of each service, as they have been initially described by domain experts in the form of scenario specifications.

4.1 Family Reunion

The family reunion service is based on the principle of “family unity”. In 1986 the first immigration law was promulgated in Italy as a result of the large number of applications submitted by foreigners in order to be reunite with their relatives. The Law went through several changes before the current version. The latest changes have been made by the legislative decree of 3 October 2008, n. 160 and then by Law 15 July 2009 n. 94 named “Measures for public safety”.

Several participants are involved in the delivery of this service. The beneficiaries are both the foreigner, which applies for family reunion (or a patronage that acts on his/her behalf), and the family member to be reunited. The different PAs involved in the service delivery are following presented.

– The prefecture is the main driver of the process, on behalf of the Department for Civil Liberties and Immigration of the Ministry of Interior according to the geographical location of the applicant.
– The Police headquarters is in charge of public security controls and they give opinions on the feasibility of the application.
– The Italian authorities abroad (consulate or embassy) are responsible for verifying the subjective requirements.
– The Ministry of Foreign Affairs communicates the results of the procedure to the Italian authorities located in the state of the requesting beneficiary.
– The Ministry of Finance is in charge of releasing the fiscal code for the incoming relative.

To support the process the Department for Civil Liberties and Immigration of the Ministry of Interior desing and deployed a "one stop shop" service for immigration, named SPI. All the 106 Italian prefectures can access and use the system, which permits to the beneficiaries to electronically apply and verify the status of the requests, via a secured access.

The main steps of the BP supported by the SPI are as following descibed.

1. The BP starts with a reunion application done by beneficiaries living in Italy using a downloadable software client freely available after registration.
2. The application is managed by the SPI and assigned to a prefecture that asks, for public safety constraints, to the Police Headquarters and than invites the beneficiary to the Prefecture in order to check her/his status. Both the opinions from police and Prefecture may be cause of application rejection. Otherwise in case of acceptance the Ministry of Foreign Affairs provides the go-ahead ("nulla osta").
3. After the release of the "nulla-osta" the relative that has to be reunited goes to the Italian consulate or embassy in her/his country, and proving some specific requirements asks or VISA in order to be admitted in Italy.
4. Once in Italy the foreigner must go (within 8 days) to the Prefecture in order to register his/her arrival in Italy, to receive the fiscal code, thanks to the interaction with the Ministry of Finance, and to finally obtain the residence permit.

4.2 Grant Citizenship

Grant citizenship is a service to be used by foreigners and stateless persons to ask for Italian citizenship. The first regulation is the Law of 13 June 1912, n. 555 implementing the concept of family relationships assigning a position of absolute pre-eminence of the husband respect to his wife, at that time commonly recognized. After several law evolutions we can state as following. The primary mode of acquisition of citizenship is by birth. With the law n. 91 5/2/1992 is upheld the principle of "ius sanguinis". At the same time, taking into account the strong migration occurred in our country, people can obtain Italian citizenship for marriage or after long residence.

Several participants are involved in such service. The beneficiary is the foreigner which applies for Italian citizenship and the providers are the different Public Administrations involved in the service delivery as following.

- The Prefecture, on behalf of the Department for Civil Liberties and Immigration of the Ministry of Interior according to the geographical location of the request, is the main actor and drives the process, receiving the request, checking the requirements and giving the opinion.
- The Ministry of Interior receives electronically the request and the documentation, checks them, valuates the instance and took the final decision;
- The Municipality officiates to the new citizen sworn;
- The Ministry of Foreign Affair, Police headquarters, Ministry of Justice and public security offices such as Information Agency and External Security (AISE), Information Agency and Internal Security (AISI), give their opinions on the application.

In order to support the process the Department for Civil Liberties and Immigration of the Ministry of Interior decided to develop an electronic system, named SICITT. It is suitable to manage requests and documentations for granting citizenship. SICITT satisfies the needs of the Ministry of Interior to communicate with other offices involved in the process of grant citizenship mainly to obtain the opinions. It is in use in all the Prefectures and in almost every police-headquarters.

The main steps of the BP are described in the following.

1. The process starts with a request done by the foreigner by ordinary mail or delivered by hand to the Prefecture. The SICITT foresees that an employee uploads the request.
2. Document verification is the second step, and the following conditions could occur.
   a) The prefecture asks to complete the documentation in the case some documents are missing. Then the applicant has to produce and delivery the required documents to the Prefecture, otherwise the citizenship office begins the procedure for instance rejection.
   b) The prefecture notifies the begin of the rejected procedure if some requirements are not satisfied. In 30 days the applicant has to solve such condition otherwise the request will be classified as inadmissible.
3. On the other side when the documentation is complete and all the requirements are satisfied the following steps are completed.
   - The request inserted in SICITT becomes visible to the police-headquarters that checks the lack of impediments, and then expresses an opinion. If the Prefecture does not receive the police-headquarters opinion in 6 months, it solicits the office.
   - Only after receiving the opinion of the police-headquarters, the Prefecture sends its opinion to the Ministry of Interior. Contemporary to the receipt of the application to the Ministry of Interior, the SICITT automatically sends a request of information to other offices: AISE, AISI, Ministry of Justice (criminal record), MAE and to the department of the anticrime police.
   - Only after receiving all the opinions, the Ministry of Interior verifies the instance and it can decide to:
     a) Ask for an integration of the documents;
     b) Start the procedure for the rejection of the instance;
     c) Confirm grant citizenship.

     Any final decision is sent to the Prefecture that is in charge to notify the applicant about the decision.
   - In case of confirmation, the Prefecture asks to the municipality to call the applicant for the oath. Only after applicant sworn the process is closed.

4.3 Bouncer Registration

The bouncer is a person employed by a cinema, recreation ground, nightclub or similar establishment to prevent troublemakers from entering or to reject them from the premises. In Italy, a national registry has been created according to the Ministry of Interior decree of 6 October 2009 and all the Italian bouncers have to satisfy specific requirements and then be registered in the registry.

Several participants are involved in the provisioning of such a service. The beneficiaries are the managers of public place or vigilance institute that do the request, and the bouncer who will be registered in the list. The providers are the different Public Administrations involved in the service delivery as following.

- Prefecture, on behalf of the Department of Public Security of the Ministry of Interior according to the geographical location of the place, has to receive the request and decides for granting or rejecting decree;
- The Police headquarters and several police departments such as Police anti-crime, General Investigation division and Special Operation (Italian acronym DIGOS) that give their opinions.

To guarantee the process the Department of Public Security of the Ministry of Interior is developing an application, named BTF, to electronically manage the requests of inscription in the bouncers registry. Up to now the BTF is going to be used by all the Prefectures and the police-headquarters, but it is expected that in a second phase it will support a fully interactive service.

1. The process starts with a request delivered by hand or by ordinary mail, from a manager of a public place or of a vigilance institute, to the Prefecture in charge to manage it. The request is successively uploaded into the BTF manually by the PA employee.
2. The Prefecture proceeds with the documents verification, it may happen that the documentation is incomplete. In this case it asks for integration to the applicant.
3. When the documentation is complete, the Prefecture analyzes it and then waits for the opinion from the police-headquarters that has to come within two weeks. If the Prefecture does not receive the opinion, it has to solicit the police-headquarters.
4. Before giving the opinion, the police-headquarters asks to other police offices, Police anti-crime and DIGOS, for receiving more information about the bouncer.
5. After receiving all the opinions from all the police-headquarters, the Prefecture decides the instance. If it is positive the inscription in the list of bouncers is authorized, otherwise the request is rejected.

4.4 Process Modeling

For each process we developed a BPMN 2.0 specification both conversation and collaboration diagrams. We do not report the diagrams here given their graphical complexity and consequently required space. Neverthless to give an idea of their complexity Table 1 reports the number of instances which are included in the different BPs for each different class of BPMN 2.0 constructs. It is easy to imagine that the high complexity of such BPs hinders their manual manipulation. Instead it is necessary to develop automatic techniques permitting to verify if unwanted situations (deadlock and livelock in our case) could emerge.

Table 1. Model Complexity and Experimental Results

Service	Pools	Activities	Message Flow	Execution Time (min)	Nodes in the Tree	Dead.	Live.	Good traces
Family reunion	8	131	36	30	253	40	8	0
Grant Citizen.	11	168	62	17	123	18	21	15
Buoncer Reg.	6	40	17	6	187	25	24	1

5 Experimentation

As for any approach using verification techniques it is important to check whether the state explosion phenomenon could hinder its applicability to real case studies. In our case we experimented with the three real processes discussed in the previous section. For all the considered processes relatively small state sets were generated. In particular, the experiments we have conducted using a desktop PC equipped with a Core 2 Duo 2,20GHz and 4GB RAM, have highlighted that defined BP can be checked with respect to the properties included in the framework in less than 30 minutes, for the most complex BP scenario. Moreover the most complex BP generated a state space of around 253 nodes in the tree for a total of around possible 48 executions paths explored. This data seems to support the idea that in the current status (i.e., complexity of BP in the e-government domain, mapping we have defined and quality properties to be checked) the approach is applicable in real scenarios and can be a useful support for BP designers. In Table 1 we report also the data resulting from the conducted experiments.

The application of our approach and has been really useful to avoid design error resulting from the complexity of the considered PA scenarios. In particular we were able to highlight several issues with the defined BP with respect to messages exchange.

As could be expected the main issues are hidden within exceptional behaviours. An interesting result refers to the fact that similar "bad traces" could be observed within different BP specifications. This could lead to the identification of a list of "risky" interactions so to provide indications for improving future BP modeling. In particular, in all the different BP we could detect livelock conditions when a document is required and it is not properly compiled, so to need further integrations. This was the case in the Bouncer Registration service. An example of deadlock occurs when there is an exclusive gateway in which at least two of its output flow will converge, after several steps, into a parallel gateway. This means that the parallel gateway does not start until all its input flows are consumed. This can never happen so a deadlock occurs. This was the case in the grant citizenship process model with reference to the request of the nulla osta from the prefecture.

With reference to specific cases we can refer to the following scenarios. In the Bouncer Registration process model a deadlock occur during the first step involving the PA manager. In fact it can happen that the PA manager waits for a document from the prefecture that will never arrive since the prefecture already has a version of the document and it proceeds without considering the PA Manager. This deadlock does not have a huge impact within a human driven scenario where the civil servant will adapt to the situation. Obviously different is the case of a scenario driven and supported by IT systems. In most of the cases the PA Manager has to provide automatically the document and the BP results in resource starvation. In the grant citizenship process model an example of deadlock occurs when livelock is observed in one of the involved pools. It happens in the last step when the municipality waits for a call from the prefecture. In this case if the request is not approved, due to document mismatching, it happens a livelock that also results in the deadlock of the municipality that still waits for the prefecture call.

6 Conclusions and Future Work

Public Administration Business Processes are highly complex and typically foresee the interaction of many different stakeholders. Given the continuous shift from human driven processes to mainly electronic driven processes, we are in urgent need of techniques for modeling and analyzing PA BPs. In this paper we presented an approach for checking deadlock and livelock of complex BP modeled using BPMN 2.0. The approach has been experimented with three complex scenario with interesting results. In particular, it is evident that just the effort of graphically modeling a business process can help domain experts to identify unwanted interactions. Nevertheless BPs enacted within the PA are so complex that their manual manipulation is not a suitable option. The verification algorithm we derived helped in identifying many dangerous traces.

In the future we intend to apply the approach to many other BP and to consider data set during the verification phase. At the same time we intend to explore if commonalities are available among the various BPs leading to similar unwanted traces. In this way we could derive a list of BP "anti-pattern" which can help the developer of BPs for PA.

References

1. European Commission: European Interoperability Framework (EIF) Towards Interoperability for European Public Services (2011)
2. Commission: 5th ministerial egovernment conference. Ministerial Declaration on eGovernment (2009)
3. Commission, E.: Communication from the commission to the european parliament, the council, the european economic and social committee of the region. a digital agenda for europe. Technical Report COM (2010) 245 final, European Union, Brussels (2010)
4. Niehaves, B., Plattfaut, R.: Collaborative business process management: status quo and quo vadis. Business Process Management Journal 17(3), 384–402 (2011)

5. Houy, C., Fettke, P., Loos, P., van der Aalst, W.M.P., Krogstie, J.: BPM-in-the-Large – Towards a Higher Level of Abstraction in Business Process Management. In: Janssen, M., Lamersdorf, W., Pries-Heje, J., Rosemann, M. (eds.) EGES/GISP 2010. IFIP AICT, vol. 334, pp. 233–244. Springer, Heidelberg (2010)
6. Lindsay, A., Downs, D., Lunn, K.: Business process – attempts to find a definition. Information and Software Technology 45, 1015–1019 (2003)
7. Roser, S., Bauer, B.: A categorization of collaborative business process modeling techniques. In: Seventh IEEE International Conference on E-Commerce Technology Workshops, pp. 43–51. IEEE, Los Alamitos (2005)
8. Object Management Group: Business Process Model And Notation (BPMN) Version 2.0 (2011)
9. Mendling, J.: Metrics for process models: empirical foundations of verification, error prediction, and guidelines for correctness. Springer (2008)
10. van der Aalst, W.M.P., ter Hofstede, A.H.M.: YAWL: yet another workflow language. Information Systems 30(4), 245–275 (2005)
11. Davies, J., Janowski, T., Ojo, A.K., Shukla, A.: Technological foundations of electronic governance. In: Janowski, T., Pardo, T.A. (eds.) ICEGOV. ACM International Conference Proceeding Series, vol. 232, pp. 5–11. ACM (2007)
12. Clarke, E.M., Grumberg, O., Peled, D.A.: Model Checking. Hardcover (2000)
13. Leroux, J.: Vector Addition System Reachability Problem: A Short Self-contained Proof. In: Dediu, A.-H., Inenaga, S., Martín-Vide, C. (eds.) LATA 2011. LNCS, vol. 6638, pp. 41–64. Springer, Heidelberg (2011)
14. McMillan, K.: Symbolic model checking: an approach to the state explosion problem. Technical report, DTIC Document (1992)
15. Hack, M.: Decidability questions for Petri nets. Garland Publishing, New York (1975)
16. Esparza, J., Heljanko, K.: Unfoldings: a partial-order approach to model checking. Springer, New York (2008)
17. Karp, R., Miller, R.: Parallel program schemata. Journal of Computer and System Sciences 3(2), 145–195 (1969)