TOWARDS DYNAMIC COMPOSITION OF E-GOVERNMENT SERVICES

A Policy-based Approach

Ivo J. G. dos Santos[1], Edmundo R. M. Madeira[1] and Volker Tschammer[2]
[1]Institute of Computing, University of Campinas, PO Box 6176, 13083-970, Campinas, SP, Brazil; [2]Fraunhofer FOKUS, Kaiserin-Augusta-Alle 31, 10589 Berlin, Germany

Abstract: The use of Information and Communication Technologies in governmental process and services, often known as e-Government, has gained momentum over the last decade. The demands for the on-line delivery of each time more complex and citizen-centric services and also the need for enabling citizen participation in governmental processes and decisions have created a series of technological challenges. If, on one hand, Service Oriented Architecture (SOA) appears as a natural and direct solution for problems like heterogeneity, on the other, issues like how to deal with the dynamism of the processes, the autonomy of the different entities involved and the privacy of data being exchanged still must be treated. We present in this article the first steps towards an effective solution to dynamically compose e-government services. These compositions are mediated through policies which provide different levels of autonomy and privacy in the involved interactions. Semantics are used to help building up the compositions, which are made effective through techniques like Orchestration, Choreography or a combination of both.

Key words: e-Government; Collaboration; Web Services; Dynamic Composition; Autonomy and Privacy Policies.

1. INTRODUCTION

The Information and Communication Technologies (ICTs) are being applied vigorously by governmental offices at national, regional and local levels around the world[1], a phenomenon often named Electronic Government (e-Government). The Online Service Delivery is perhaps the most common manifestation of this phenomenon. Nevertheless, citizen participation in government decisions through electronic means (e-Democracy) is also gaining momentum.

When it comes to delivering more complex services (which involve more than one entity) a series of problems arise. The first one is the implicit heterogeneity of the information systems in each one of those entities and therefore a challenge is to find a way to integrate them (or compose the services they provide to deliver a new one) without the need of doing big changes in the already running systems. Besides that, issues like the autonomy of the entities involved in the service provision and the privacy of the information being exchanged among different partners play important roles to make the integration a reality.

The SOA (Service Oriented Architecture) appears as a powerful choice to solve the integration problems mentioned before. Its concept of applications based on (Web) services and their compositions, regardless of how those services are implemented, helps breaking the first integration barrier: the technological heterogeneity. But it also creates new challenges, like finding effective ways to describe those services, to compose them dynamically and also to mediate (or not) their interactions.

We show in this article the first steps towards an effective solution to dynamically compose e-Government services. We rely on SOA as a technological base and propose the use of policies to mediate the compositions. These policies are intended to provide different levels of autonomy and privacy inside the composite services. The contribution we present in this work is in the context of the development project of a Collaborative e-Government Platform (*CoGPlat*), a middleware which intends to offer a set of services and facilities based on e-Governance and e-Democracy premises.

This article is organized as follows: in Section 2 we present as background an overview of the main concepts and technologies related to our work; in Section 3 a brief discussion on the state-of-the-art of e-Government projects and also on Cross-organizational Systems Collaboration research is shown; in Section 4 we introduce and discuss our proposal for the transparent composition of e-Government services; in Section 5 we present the final considerations, future steps of our project and also suggest extensions to our work.

2. BACKGROUND

2.1 E-Government: e-Governance and e-Democracy

The term Electronic Government (*e-Government*), as an expression, was coined after the example of Electronic Commerce. In spite of being a relatively recent expression, *e-Government* designates a field of activity that has been with us for several decades and which has attained a high level of penetration in many countries[2].

What has been observed over the recent years is a shift on the broadness of the *e-Government* concept. The ideas inside *e-Governance* and *e-Democracy* are to some extent promising big changes in public administration. The demand now is not only simply delivering a service on-line. It is to deliver complex and new services, which are all citizen-centric. Another important demand is related to the improvement of citizen's participation in governmental processes and decisions so that the governments' transparency and legitimacy are enforced. In order to fulfill these new demands, a lot of research has been done over the recent years (see Section 3) but many challenges are still to be faced, not only in the technological field, but also in the political and social aspects.

2.2 Service-oriented Architecture

Service-oriented Architecture (SOA) is a component model that inter-relates the different functional units of an application (called *services*) through well-defined interfaces and contracts between these services. These interfaces are defined in a neutral manner that should be independent of the hardware platform, the operating system, and the programming language the service is implemented in. This allows services, built on a variety of such systems, to interact with each other in a uniform and universal manner [3,4].

On the Internet, the *Web Services* technology represents a manifestation of the SOA. A *Web Service* can be defined as an application which is made public through the publication of an interface (or a port) - note that the way the application is implemented is not important at all to a service client. The service interface is described and accessed through a set of Internet standards and protocols, like XML (eXtended Markup Language), HTTP (HyperText Transfer Protocol), SOAP (Simple Object Access Protocol) and WSDL (Web Services Description Language) [5].

2.2.1 Service Compositions

There are some scenarios (in fact most of them) in which the access to a single service is not enough to fulfill a goal. In order, for instance, to implement a business process or to solve a scientific problem using a SOA approach, it is usually necessary to compose various services. The fact that these compositions may themselves become new services makes composition in SOA a recursive operation[6].

There are two composition techniques usually considered when composing *Web Services*: orchestration and choreography. There isn't yet a common sense regarding these concepts, but to the context of our work, we will adopt the following definitions[7]:

- ***Orchestration***: A service composition described as an *orchestration* describes all interactions that are part of a process in terms of sequence of activities, conditional events, etc. It is very similar to a workflow description on a traditional workflow system. The viewpoint is often centralized and the description is executed by an orchestration engine;
- ***Choreography***: A service composition described as *choreography* is more collaborative and less centralized in nature. Only the public message exchanges are considered relevant. Differently from Orchestration, there is not an entity that has a global view/control of the composition. The *choreography* definitions are used usually as guidelines/protocols during the composition development time. They may also be used as validation rules during the composition execution time.

In "real world" scenarios usually both approaches must be considered. In terms of specifications, the most relevant to the context of our work are BPEL4WS[8] and WS-CDL[9]. The BPEL4WS (Business Process Execution Language for Web Services), or simply BPEL, defines a language based on XML that describes the control logic required to coordinate the participant services on a process flow - BPEL is, essentially, a layer over WSDL. BPEL defines both Abstract and Executable business processes. The WS-CDL, or simply CDL, represents an on-going effort being held by the W3C (World Wide Web Consortium) to establish a choreography standard language. CDL is also XML-based and describes peer-to-peer collaborations of Web Services by defining their common and complementary observable behavior. WS-CDL does not treat executable processes, but only the choreography aspects of a composition, and therefore can be used as a compliment to BPEL and other composition specifications.

2.2.2 Semantic Web and Services

A series of efforts towards the creation of the Semantic Web are gaining momentum[10]. From a high-level perspective, one of the major goals of the Semantic Web is to shape information in an unambiguous, machine-processable way, thus enabling a better exchange of information both between humans and computers and among computers[11]. The major effort of the Semantic Web community has been the release of expressive languages and formalisms to describe information, such as OWL[12] (Web Ontology Language) and RDF[13] (Resource Description Framework). In terms of Web services, a computer-interpretable description of the service (and the means by which it is accessed) must be provided. OWL-S[14] is a language that tries to define the ontology necessary to provide such descriptions of services on the Semantic Web, promising to enable the automatic Web service discovery, invocation, composition and also execution monitoring.

3. RELATED WORK

3.1 E-Government

Next we briefly present some projects in the e-Government area which, as our work, study ways to enhance the provision of e-Government services, and therefore complement our proposal.

EU-Projects. The European Union (EU) has over the recent years paid a lot of attention to the challenges of e-Government and a series of research projects have been sponsored by its commissions. Some of these projects are briefly presented next.

The *TerreGov* (*Impact of e-Government on Territorial Government Service*) is an on-going project which addresses the issue of interoperability of e-Government services[15]. Its first results include a series of studies on the state-of-the-art in e-Government, on the requirements for new e-Government applications and also on the technological alternatives that could be used to implement the desired interoperability (where *Web Services* and also *semantics* play an important role).

The *eMayor* is a collaboration project between municipalities of four countries of the EU, several universities and firms providing technology[16]. It is supported by the European Commission's Research and Development Department. It promises to develop and implement an open, secure, and affordable e-Government platform for small and medium European public organizations. It intends to support the secure communication of municipalities amongst themselves, businesses and citizens.

Another relevant on-going project, *eGOIA* (*Electronic Government Innovation and Access*), aims to implement a demonstration system supporting the access of citizens, through the Internet, to integrated public e-Government services[17]. Technically the project is based on two main paradigms – back-office and front-office integration. The first concentrates on a unified approach to access already existing and newly emerging government services. The second focus on providing an intuitive user-interface integrating the diverse e-Government services available.

Some important results were also presented by the already finished *PRISMA* project[18], especially in terms of providing a systematic analysis and synthesis of current and future impacts of new ICTs on government services in Europe. Within the context of e-Government and e-Democracy, six major service fields have been examined in detail: administrations; health; persons with special needs: the disabled and elderly environment; transport; and tourism.

3.2 Cross-organizational System Collaboration

The problem of cross-organizational system integration appeared first in e-Business workflow systems and is also present in the e-Government field. Schulz[19] proposes a classification of business processes into private and shared. The private processes expose interaction points where the shared processes connect to, in such a way that a business process can be part of two or more organizations. A framework to support these two categories of processes, BPFA (Business Process Framework Architecture) is also introduced. BPFA consists of a set of components that execute instances of an inter-organizational process model, extending a company's workflow infrastructure and allowing process-oriented communication among partners and customers.

Santos and Madeira[20] extended Schulz model proposing a set of policies to regulate the interactions and applied it with Dynamic Virtual Enterprises. The interaction policies adopted in our work (Section 4.3) extend these two proposals and adapt them to a Collaborative e-Government environment.

Dijkman and Dumas[21] propose a multi-viewpoint approach, based on a control-flow perspective in terms of *Petri nets*[22], to design composite services. They identify four viewpoints from which it is possible to describe the control-flow aspect of Web Services: the choreography viewpoint, the interface behavior viewpoint, the provider behavior viewpoint and the orchestration viewpoint.

4. TRANSPARENT COMPOSITION OF E-GOVERNMENT SERVICES

4.1 CoGPlat: A Collaborative e-Government platform

The proposal for transparent composition of e-Government services we present next in this section is part of the development project of a Collaborative e-Government platform called *CoGPlat*[23]. *CoGPlat*'s main goal is to support the interaction and the collaboration of governmental entities, organizations (public, private and nonprofit) and citizens in different public administration scenarios, ranging from the electronic delivery of integrated services to the support for citizen participation in government decisions. *CoGPlat* is being developed as a service-oriented middleware (not as an end-user application) in order to provide a set of general services and facilities to be used by specific applications and tools. In this article we are going to focus in one of these facilities, the *Transparent Services Unit*, which is responsible for the dynamic integration and management of public services.

4.2 The Transparent Services Unit

The *Transparent Services Unit* (*TSU*) is the *CoGPlat*'s facility responsible for composing multiple (Web) services into new ones in a transparent manner. These new services are then delivered to citizens and/or entities through applications that run over the platform. Besides composing the services, the TSU is also responsible for managing their execution, which is regulated by a set of interaction policies (Section 4.3).

In order to dynamically realize those compositions, the TSU matches the semantic descriptions of the available services to the desired functionalities of the new e-Government service. It then builds up an execution plan, which can be an orchestration, choreography or a combination of both.

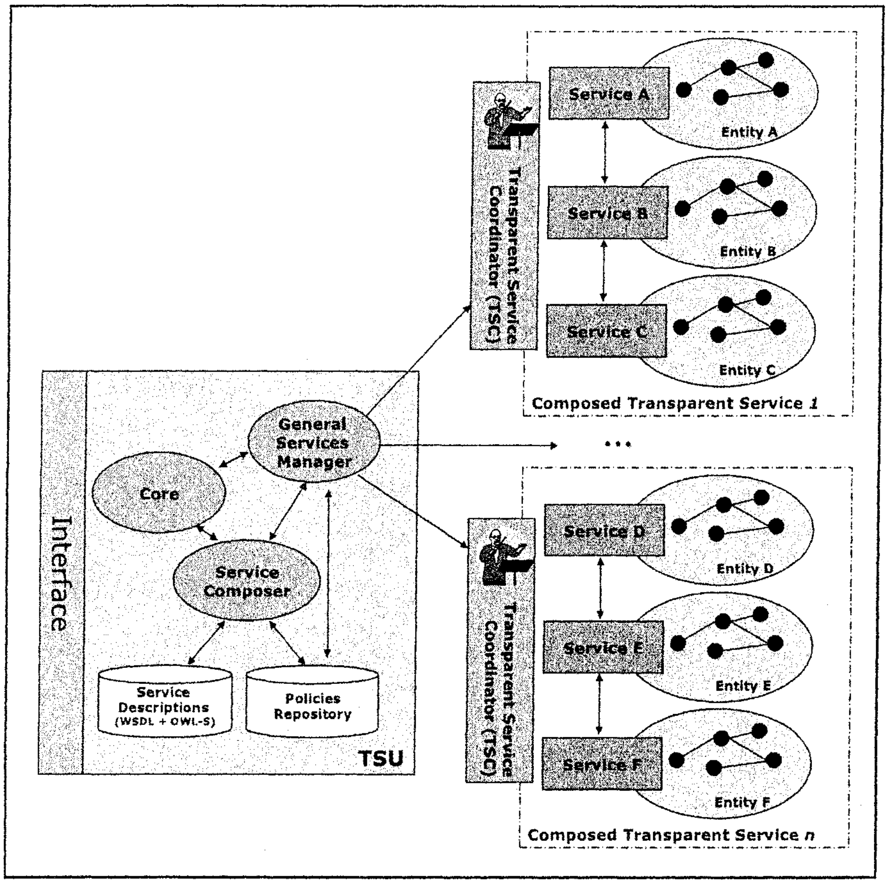

Figure 1. Transparent Services Unit

Figure 1 presents an overview of the TSU infrastructure, composed of the following elements:

- *Interface*: provides communication with the other platform units and to the client applications;
- *Core*: executes the coordination among the other TSU elements. It is also responsible for non-functional requisites such as persistency, transaction support etc;
- *Service Composer*: dynamically builds the execution plans (service compositions) by matching new service demands with the available service descriptions (WSDL + OWL-S) and also according to the existing policies;
- *General Services Manager*: coordinates the various TSCs;
- *Transparent Service Coordinator* (TSC): the TSC is responsible for managing a Transparent Service instance, coordinating the composition

and guaranteeing its correct execution, either through orchestration, choreography or their combination. There is one TSC associated with each new Transparent Service.

- *Service:* a service is considered to be an application which has a well-known interface and which provides some functionality that can be used by applications over the platform. A service in our context is always linked to an entity and can be itself a composition of services internal to this entity.
- *Entities:* are real organizations, usually representing an administrative domain. Can be government unities, private companies, NGOs etc.

4.3 Policies

When composing e-Government services, the issues of autonomy of the entities involved and of the privacy of the date being exchanged is always present. In order to enable compositions which fulfill these demands and provide different levels of autonomy and privacy among the entities participating in a Transparent Service, *CoGPlat* implements a set of Interaction Policies. These policies were previously proposed and successfully applied in a *Virtual Enterprise* scenario[20], being now adapted to the context of e-Government applications.

The Interaction Policies are classified into two categories: *Entity Autonomy Policies* (CoGPlat x Entity) and *Entity Cooperation Policies* (Entity x Entity).

Entity Autonomy Policies. Determine the level of control the platform (and therefore the applications running over it) will have over the internal stages of a service. At the moment an entity service is selected to participate in a Transparent Service, a negotiation takes place to define what level of interaction this entity wishes to have with the platform. According to the chosen level, the TSC can act in one of the following manners:

1. *Supervisor*: The TSC does not have any action on the entity inside domain. A *Supervisor* policy is sub-divided into:
   - *Consulting-only*: the TSC can only ask for status information about a service running on the entity's internal domain;
   - *Selective*: the TSC and the entity negotiate in which points of the execution plan interactions will be allowed;
   - *Participative*: the TSC can interact with all activities of the execution plan
2. *Executor*: the TSC has total control over the tasks running on the entity's internal workflow, being the responsible for composing also these tasks.

Entity Cooperation Policies. Determine the collaboration levels on the interactions between two entities that participate on the same Transparent Service instance. The following policies are available:

1. *Total Cooperation*: the two entities fully trust each other, and therefore can communicate in a decentralized peer-to-peer manner;
2. *Controlled Cooperation*: there is a pre-established set of information that should be passed to the next entity and another set that should be hidden by the TSC;
3. *Total Privacy*: there is no interaction between the entities. All information is returned to the TSC, which has access to the service execution plan and then decides what to do next, hiding from the following entity the activities and data from the previous one.

Policies and Composition. An entity is allowed to select different policies when participating on different dynamic services. In addition, a dynamic service is usually composed of multiple entities and therefore can have multiple policies. So, policies are fundamental to the dynamic construction of the execution plan and also to the correct selection of the service composition mechanisms to be used (Orchestration and/or Choreography).

As already mentioned, the TSC is the entity responsible for implementing the correct execution of the plan on the entities and for applying the interaction policies. When, for example, the composition is made up by services of an entity that controls others, an orchestration approach is more appropriate. On the other hand, when there is only collaboration among the entities (no administrative links or hierarchy and fully decentralized control), choreography is the most appropriate choice. Table 1 presents some examples of the behavior of the TSC (type of composition) according to the selected Entity Autonomy Policy.

Table 1. Examples: Autonomy Policies x Composition

	Autonomy Policy	*Composition*	*Service Control*
(a)	Supervisor :: Consulting-only	Choreography	Decentralized
(b)	Supervisor :: Selective	Orchestration + Choreography	Partially Decentralized
(c)	Supervisor :: Participative	Orchestration	Centralized on TSC
(d)	Executor	Orchestration	Centralized on TSC

In Table 1: (a) According to the Consulting-only Supervisor policy, the TSC has no intervention on the entities' services and can only participate on a choreography to validate the messages exchanged; (b) The policy here allows the TSC to interact with some of the internal services, performing a partial orchestration; (c)Here, the TSC has total control over the internal activities, and performs a full orchestration of the internal services; (d) This last policy allows the TSC to access the internal resources of the entity's workflow system. Here it can either use orchestration or even dispatch objects (mobile agents, for instance), which will have control over the local activities.

4.4 An Example

An interesting example of where a *Transparent Service* could be used is the process of obtaining an authorization to build a house near the coast, emitted by the municipality in conjunction with other authorities. Suppose that, to get this hypothetic authorization, different entities must be contacted, on a specific and pre-determined chronological sequence: (1) City Hall, (2) Civil Engineering Department, (3) Navy, (4) Fire Department and again the (5) City Hall (and suppose that steps 2 and 3 are independent and could be done in parallel). This process is usually slow because it consists basically of the transportation of documents from one entity to the other. Inside each one of the entities, an internal process is performed when the request arrives in order to make it available for the next entity. This is a typical scenario where a Composite Service could be created to handle all this process flow for the citizen. Besides the transparency offered, the process could also gain speed because each of the participating entities would have its services integrated in a seamless manner to the platform (according to the chosen autonomy and cooperation policies).

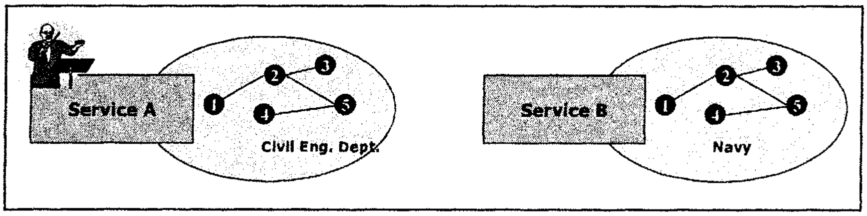

Figure 2. The Authorization Example

As seen in Figure 2, the Civil Eng. Department (administered by the City Hall, the owner of the *CoGPlat* infrastructure in this example) could select, for instance, to have a *Supervisor Participative* relationship with the TSC, letting the platform orchestrate the internal steps involved in the process. The Navy, on the other hand (and for obvious reasons), could choose the most autonomous policy (*Supervisor Consulting-Only*) so that no interference from the platform is allowed in its internal processes, preserving the privacy and integrity of its internal data.

5. FINAL CONSIDERATIONS

We present in this article the first steps towards an effective way to dynamically compose e-Government services. The contribution we present is in the context of an on-going project which intends to model and develop a Collaborative e-Government Platform (*CoGPlat*). The focus of this article is to describe and discuss one of the facilities of this platform, the *Transparent Services Units*.

Service-Oriented Architecture, *Orchestration*, *Choreography* and *Semantics* are applied to facilitate the dynamic integration of heterogeneous services delivered by different governmental and private entities.

Besides the completion of the TSU prototype, the next steps of our project include the adoption of the WS-CDL specification as a standard for the choreography of services in the whole CoGPlat platform (as BPEL already is for orchestration) and also the development of a series of other example applications (starting with Municipalities and Metropolitan Regions) to show the potentials of the infrastructure. We are also studying the possibility of adopting some formalism (like Petri-Nets or Pi-Calculus[24]) to be a standard way to describe the service compositions.

ACKNOWLEDGEMENTS

The authors would like to thank CAPES, CNPq and the WebMaps project for financial support. Additional thanks to the Fraunhofer FOKUS Institute and its ELAN Competence Center.

REFERENCES

1. G. Marchionini, H. Samet, and L. Brandt. Digital government. *Communications of the ACM*, 46(1):25-27, January 2003.
2. K. Lenk and R. Traunmüller. Electronic government: Where are we heading? In *EGOV 2002 - LNCS*, vol. 2456, pp 1-9, Springer-Verlag, 2002.
3. M.P. Papazoglou and D. Georgakopoulos. Service-oriented Computing. *Communications of the ACM*, 46(10):25-28, October 2003.
4. IBM. New to SOA and Web services. *http://www-128.ibm.com/developerworks/webservices/newto/*, July 2005
5. World Wide Web Consortium (W3C). Web Services Description Language (WSDL) Version 2.0, W3C Working Draft, *http://www.w3.org/TR/wsdl20/*, 2004.
6. F. Curbera, R. Khalaf, N. Mukhi, S. Tai, and S. Weerawarana. The next step in Web Services. *Communications of the ACM*, 46(10):29-34, October 2003.
7. C. Peltz. Web Services Orchestration and Choreography. *IEEE Computer*, 36(10):46-52, 2003.

8. BEA Systems, IBM, Microsoft, SAP AG, and Siebel Systems. Business Process Execution Language for Web Services (BPEL4WS) - Version 1.1. *http://www-128.ibm.com/developerworks/library/specification/ws-bpel/*, 2003.
9. World Wide Web Consortium (W3C). Web Services Choreography Description Language Version 1.0 (WS-CDL). *http://www.w3.org/TR/ws-cdl-10/*, Working Draft, 17 Dec 2004.
10. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5):34-43, 2001.
11. J. Barthès and C. Moulin. TERREGOV Technological State of the Art - v1, version 2, *http://www.terregov.eupm.net*, December 2004.
12. D. L. McGuinness and F. van Harmelen. OWL Web Ontology Language Overview. *http://www.w3.org/TR/owl-features/*, W3C Candidate Recommendation, August 2003.
13. World Wide Web Consortium (W3C). Resource Description Framework Primer. *http://www.w3.org/TR/rdf-primer/*, W3C Recommendation, February 2004.
14. The OWL Services Coalition. OWL-S: Semantic Markup for Web Services. White paper. *http://www.daml.org/services*, July 2004.
15. TERREGOV Project Web Site, *http://www.terregov.eupm.net*, in May 2005.
16. eMayor Project Web Site, *http://www.emayor.org*, in May 2005.
17. eGOIA - Electronic GOvernment Innovation and Access, Technical Report n. 01, *http://www.egoia.sp.gov.br/pub/Annual-Technical-Report-eGOIA-2004-Delivered.pdf*, 2004
18. PRISMA Project Web Site, *http://www.prisma-eu.net*, in May 2005.
19. K. Schulz and M. Orlowska. Architectural issues for cross-organizational b2b interactions. In *ICDCSW '01: Proceedings of the 21st International Conference on Distributed Computing Systems*, page 79, USA, IEEE Computer Society, 2001.
20. I.J.G. Santos and E.R.M. Madeira. Vm-Flow: Using web services orchestration and choreography to implement a policy-based virtual marketplace. In *Proceedings of the World Computer Congress 2004 - 4th IFIP Conference on e-Commerce, e-Business, and e-Government*, vol. 9, pp. 265-285, Toulouse, France, Kluwer Academic Publishers, August 2004.
21. R. Dijkman and M. Dumas. Service-oriented design: A multi-viewpoint approach. *International Journal of Cooperative Information Systems (IJCIS)*, 13(4):337-368, 2004.
22. W.M.P. van der Aalst. The application of Petri nets to workflow management. *The Journal of Circuits, Systems and Computers*, 8(1):21-66, 1998.
23. I.J.G. Santos and E.R.M. Madeira. Cogplat: Using composition to enable collaborative e-government services. In *EU-LAT Workshop on e-Government and e-Democracy, Vol. 8 of e-Government and e-Democracy: Progress and Challenges*, pp. 17-27, Santiago, Chile, May 2004.
24. R. Milner. Communicating and Mobile Systems: The pi-Calculus. Cambridge University Press, 1999.