A SEMIOTIC APPROACH FOR FLEXIBLE E-GOVERNMENT SERVICE ORIENTED SYSTEMS

Rodrigo Bonacin

CenPRA, MCT, Rodovia Dom Pedro I, km 143,6, 13069-901, Campinas, São Paulo, Brazil

M. Cecilia C. Baranauskas

Institute of Computing, Unicamp, Caixa Postal 6176, 13083 970, Campinas, São Paulo, Brazil

Thiago Medeiros dos Santos

CESET, Unicamp, Rua Paschoal Marmo, 1888, Limeira, São Paulo, Brazil

Keywords: Organisational Semiotics, Tailoring, Service Oriented Architecture, e-Government.

Abstract: E-Gov is a multidisciplinary field which addresses many issues ranging from the social sciences to the technological ones. One of the big challenges is the underlying complexity to elicit and model requirements. In practice it is quite hard to encompass the requirements of all citizens or organisations involved in the project. To deal with this challenge we propose a flexible distributed systems approach, which allies tailoring concepts to Organisational Semiotics methods in a SOA based architecture. It is based on two Organisational Semiotics methods: the Semantic Analysis, which delivers a stable ontology of the context, and the Norm Analysis, that can be used to specify the volatile individual and collective requirements. The paper shows how norms changes in high level interfaces can have effect on different components of the software architecture. The architecture is experimented in a proof of concept for an e-Gov project.

1 INTRODUCTION

E-Gov systems represent a typical scenario in which the integration of human factors in a distributed system is a critical condition for the interaction and communication between the involved parts. While it is a multidisciplinary field and addresses many issues ranging from the social sciences to the technological ones, a lot of research efforts have been put in the technology. The use of interoperability techniques is especially important in e-Gov. Several projects are adopting Service-Oriented Architectures (SOA) solutions to deal with the interoperability problems.

Besides providing alternative ways of accessing public services, e-Gov projects are also considered an opportunity to promote the digital inclusion of the less favoured people and societies. This diversity of users and needs certainly represents one of the big challenges of the Human-Computer Interaction and Requirements Engineering fields for the integration

of human factors in distributed systems. Citizens with different abilities, educational levels, physical conditions and preferences require flexible interfaces and services.

The development of multiple end-user interfaces and back-office services that implement all citizens' needs and preferences is certainly impracticable. To deal with this challenge, in this work we propose an approach to create flexible distributed systems, which aims to make the software adaptation and personalization a smoother process.

By using proper tools, specialists in social inclusion, domain specialists and the citizens could themselves adapt and maintain the system, at least in part, according to volatile individual and collective requirements.

Changes in features of distributed system, such as the related to the business process execution, quality of service and security, can cause collective effects on the system use. In resource constrained environments it is necessary to define who can adapt each aspect of the system (i.e. users, domain

specialists, managers, and designers), when this adaptation can be done, and where and how it is done. In order to answer these questions it is necessary to analyse the social aspects that result in the definition of responsibilities, duties and permissions of the agents over the system.

We draw on the idea of Tailoring and on the theoretical and methodological background of Organisational Semiotics to design for flexibility. "Tailoring" can be understood as "the activity of modifying a computer application within the context of its use" (Kahler et al., 2000, p. 1). The "Organisational Semiotics" theories and methods help us to comprehend the organised behaviour by studying the use of signs in the real world.

The paper is organised as follows: Section 2 discusses the application of OS concepts and methods, Section 3 discusses an alternative to provide flexible e-Gov systems, Section 4 presents the proposed architecture, methods and tools, Section 5 discusses a proof of concept, and Session 6 concludes.

2 ORGANISATIONAL SEMIOTICS CONCEPTS AND METHODS IN DISTRIBUTED SYSTEM DESIGN

Semiotics, the ancient doctrine of signs, leads us to a precise understanding of information as various properties of signs. Anything standing for another thing or used to signify something else (Peirce, 1931-1958) is an example of a sign: words, sentences, traffic lights, diagrams, a wave of hand or a facial expression.

From a semiotic point of view, people can communicate across multiples organisations, cities, or countries by sharing signs (socially constructed) through distributed systems support. By studying the use of signs in the organised behaviour, Organisational Semiotics (Liu, 2000) theory and methods can support us in understanding and modelling social aspects of distributed systems. In the e-Gov, entities have to collaborate by using computational system in order to deliver services to the population. Semantic, Pragmatic and Social aspects have influence of in the information sharing and reuse; methods such Semantic and Norm analysis (Stamper et al., 1988) can represent the domain knowledge in order to promote the interoperability (Fu and Liu, 2006).

In order to reduce the complexity of constructing tailorable systems, we propose the separation of the system into two parts (Bonacin et al., 2004): a "static" part that can be adapted or modified by programmers, and a "dynamic" part that can be adapted or modified by end-users (or domain specialists). We argue that the division of the "static" and the "dynamic" parts should be mostly based on social aspects surrounding the system usage domain.

The architecture and tools to support the construction of "tailorable" systems is based on two methods of Organisational Semiotics (Liu, 2000). The first method adopted is the Semantic Analysis, which delineates the area of concern of an organisation and identifies the basic patterns of behaviour (affordances) of their agents. The Semantic Analysis produces a stable ontology of the context. This ontology describes the semantic of the signs used in a distributed system.

The second method is the Norm Analysis, which describes the relationships between an intentional use of signs and the resulting behaviour of responsible agents in a social context; they also describe the beliefs, expectations, commitments, contract, law, culture, as well as business. Besides the description of the agents' responsibilities in the organisation, Norm Analysis can also be used to analyse the responsibilities of maintaining, adapting and personalising the system features.

The dynamic part of a "tailorable" system can be derived from norm specifications; when a norm is changed or adapted in a specific situation, the system is also modified. The norms specify what can be changed and also who is responsible and how the dynamic part can be changed.

3 PROVIDING AN E-GOVERNMENT FLEXIBLE SOLUTION

Information sharing, universal access, confidentiality, interoperability, open architecture and standardization are usually addressed in e-Gov projects (Marchionini et al. 2003). Governments from the local to the national administration levels have developed and deployed Information Systems on the Web and are widely using Information Technology (IT) in order to promote access to information, completion of transaction services and citizen participation (Marchionini et al. 2003). Service-Oriented Architecture (SOA) (Newcomer and Lomow, 2004) has been increasingly used as an alternative to deal with interoperability issues due to the very heterogeneous back-office (internal IT infrastructure) of the public area, and to provide multi-channel access to e-Gov services.

We can understand SOA not only from its technological features, which emphasize interoperability, reuse, flexibility, among others, but also as a wider term, which includes concepts, policies and practices used to provide relevant services for their customers (Sprott and Wilkes, 2004). According to this point of view, it is necessary to identify and design relevant and adapted services from the customer perspective in order to develop valuable systems.

Nowadays, Web Services is the most popular technology to implement SOA. In many cases, e-Gov interoperability patterns give a special attention to Web Services patterns. In this paper we consider the Web Services technology and the concepts behind them as our implementation platform.

We aim to provide architecture and tools for supporting the construction of "tailorable" SOA based system. These tools adopt open standards and web technology to promote the flexibility in SOA based systems. The tools support the adaptation of the front-office and back-office systems based on the modification in norms modelling.

Regarding the front-office, the system can be adapted for example to multiple devices, necessities, and preferences through the specification of norms that are present in each context of use. Each citizen could adapt the interface according to their needs; specialists in digital inclusion could customize the interface to facilitate the system accesses, and business specialists and/or designers could change the front-office according to the business changes.

Norms are used to specify aspects related to the system usage as well as business. In many cases norms can be linked to end-user interface elements that represent actions that should be done by the agents; therefore, if the norm changes the system will also change. The norms also specify the access to the interface that provides tailoring and who has the authority to change each group of norms.

Regarding the back-office, services orchestrations, compositions, qualities, and network settings can be defined by modifications in norms specifications. For example, norms can specify who can approve a requirement in a business process. A business specialist could change this norm.

4 ARCHITECTURE AND TOOLS

Figure 1 shows an overview of the architecture for flexible e-Gov systems. This architecture provides the platform independency between front-office and back-office systems by using *Web Services* technology. We can have multiple tailorable end-user interfaces implemented using different
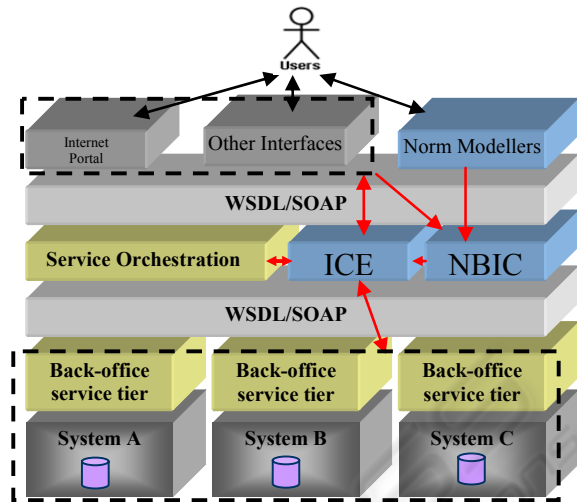

Figure 1: The Architecture Overview.

technologies, accessing heterogeneous back-office systems. These interfaces are individually or simultaneous affected by the norms.

The ICE (Interface Configuration Environment) acts changing the front-office, the services orchestration, and back-office systems. The ICE receive context information from the architecture components, evaluate the norms related to context by using an inference machine and return to the components an action plan with the changes do be done.

The NBIC (Norm Based Interface Configurator) receives the norm specification in Deontic logic, manage the norms persistence, and also transform it in a platform specific language that can be interpreted by an inference machine on ICE. The norms can be specified or changed by using norm modellers, programs that aims to provide high level interfaces for the users. In the architecture, a norm modeller is any application which specifies or changes norms through the NBIC.

According to Figure 1, Norms can also be captured from the system usage and passed to NBIC; the front-office interface can capture activities that result on parameters that inform some norms related to the system usage. In these interfaces, norms descriptions are not explicitly specified by the users.

4.1 Example of Usage

Figure 2 shows the interaction between a norm modeller, the NBIC, the ICE, and a hypothetical e-Gov system. The following simplified scenario illustrates how this interaction occurs:

− A business specialist specifies the following norm by using the norm modeller interface: *Whenever a citizen request a medical examination, If the responsible Physician has free schedule positions (s)he is Obliged to Confirm the Appointment to the next position;*

− The norm modeller tools include the norm in the NBIC by accessing service operations;

− The NBIC translates the norms to a platform specific language required by the inference machine, and include in the ICE database;

− The system access the ICE sending the context information, through a "perception mechanism", it could inform for example: *the logged user is the Physician responsible for the examination, and there is a request on the queue;*

− The ICE infers actions to be done by using the norms specifications and the context information. It could infer that: *Physician is Obliged to Confirm the Appointment to the next free position*;

− By using rules that link the norm (Bonacin et al., 2004) to system actions in an action plan, it can infer that: *a service that manage the schedule and an interface to confirm the appointment must be called at each logon;*

− The system receives the actions to be done by accessing the ICE, and an action mechanism interprets the action plan.
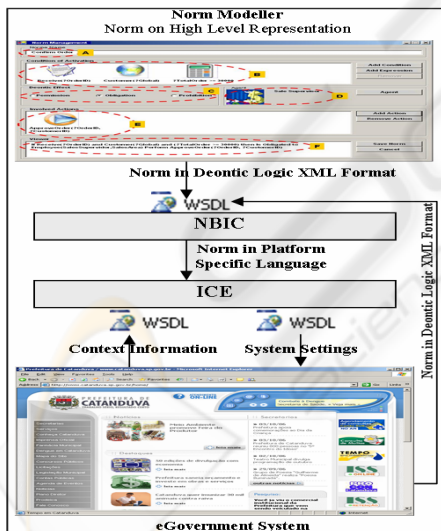

Figure 2: Example of NBIC and ICE in use.

4.2 Tailoring Multiple Architectural Components

In the architecture, the tailoring activities can result in changes at multiples components. The next sessions briefly show alternatives to the construction of tailorable components.

4.2.1 Tailoring the Front-Office Interface

In order to facilitate the construction of the perception and action mechanisms and to improve the productivity of the front-office development, we have developed the Tailoring Framework (TF) (Figure 3), which provides developers with an extension of the Apache Struts Framework (Husted et al., 2006), a largely used framework for web development with Java technology.

The static part of the interface can be implemented as a regular Struts application, and the dynamic part is specified through new tags to be included into JSP (Java Server Pages) source code that access XML based files with the adaptable and extensible content. The XML files content substitute the parts indicated as tailorable during the execution time. It will produce code that can be interpreted by the web browsers such as: *html* and script languages. The information about the application context is captured by the TF and inserted into ICE in a transparent way.

Based on the norms specification, the inference machine infers actions to be done. The actions are captured by the framework and included in session variables. Additional tags access the session variables with the action plan and specify where changes in the interface will occur and the content to be presented according to the XML files. The XML files specify objects and components that are modified according to the inferred action. The tags also encapsulate the complexity of representing norms and usage information.
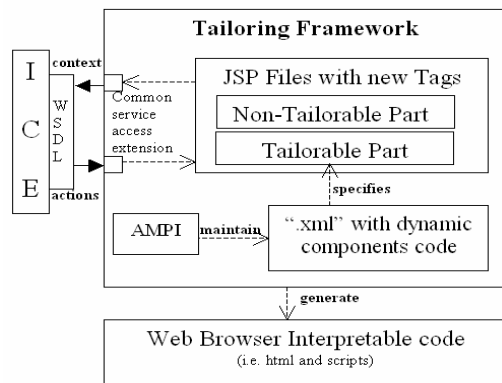

Figure 3: The Tailoring Framework.

An additional tool, the AMPI tool, can be used by an interface specialist to manipulate the XML files. As the interface is defined by the actions inferred by norms interpretation, the end-user

application interface provides to them the opportunity to change the system by changing the norms specification.

4.2.2 Tailoring the Service Orchestrations

In SOA, services can be orchestrated and composed in order to constitute complex services according to business requirements. Nowadays, BPEL (Business Process Execution Language) (Thatte et al., 2003) engines are widely used for this purpose.

In the architecture, part of the process is defined using BPEL and part using norms. As a final result, when a norm is changed the effects are reflected in the orchestration. Figure 4 shows the ICE interaction with business processes execution. In this alternative, several processes of different engines instances can invoke the ICE as a partner link.
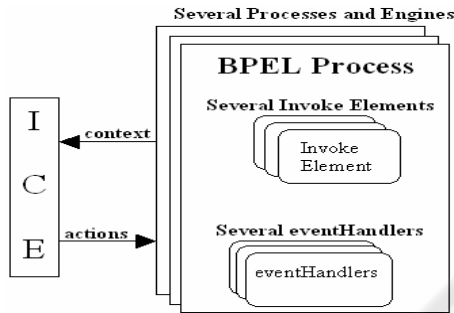

Figure 4: ICE and Business Processes Execution.

The BPEL invocation element can be distributed over multiple points of each process specification in order to inform the usage aspects from different parts of the architecture. The norms are not necessarily attached to a specific business process, but some norms are valid in a wider context, since they are specified into ICE.

The norms evaluate the context as a whole and the actions can have effects on many processes. An action plan is received by each process, and BPEL eventHandlers deal with each action to be done. Process activities and decisions are associated to each action.

4.2.3 Tailoring the Back-Office

According to Figure 5, for each service exported by the back-office, an additional operation was included that implements an action mechanism, which evaluates the action plan and does the appropriated changes. The back-office service tier links the actions to the back-office elements.

The main difficulties in the back-office tier adaptation are the presence of legacy systems with low level of flexibility. The actions and

consequently the effect of the norm changes on back-office are restricted by the legacy system flexibility. However, the flexible part of the systems (even if small), and the new flexible back-office services can be changed by norms if the appropriated mapping is done. For each system, a specific strategy should be applied according to the internal architecture of the back office system.
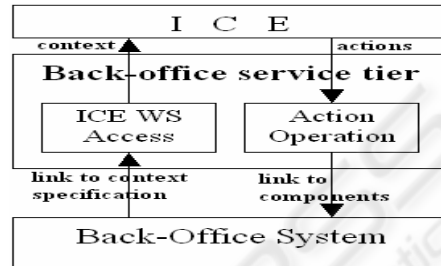

Figure 5: Back-Office tailoring.

Additionally the back-office system should also provide context information for the ICE. According to Figure 5, a service tier encapsulates the access and provides the data structure required by ICE.

5 THE APPLICATION OF THE PROPOSED ARCHITECTURE

The architecture was applied to a proof of concept in a web site of an e-Gov project that we are conducting (Bonacin et. al., 2006). The town of Catanduva is based on agricultural activities, and it is starting an industrialization process. The majority of Catanduva citizens has few or no access to IT infrastructure.

We were contacted to provide a solution to allow the citizens to access the government services, but understanding "access" in a broad sense, i.e., we were called to design a technically and socially accessible way of allowing the citizens to effectively interact with the system. The architecture was applied in a minimalist web site design where the citizens could participate of the project given opinions and including questions about issues related to the main public areas such as: education, healthcare, tributes, sport and transport.

We have explored three scenarios: the access by people with subnormal vision, the access by people with difficulty to use the mouse, and a business requirement change that results in the necessity to audit the site.

Although this web site is not big enough to stress all the architectural features, it pointed out some

issues to be addressed in the practical application in real use:

- Performance and scalability issues. The ICE acts as a core element of the architecture, the application requests the ICE many times even in simple tasks. This issue can be addressed by using replication and parallel processing;
- Security issues. Some elements of the architecture and tools present security faults, for example: there is not an authentication of norms modellers;
- Norms complexity and versioning. In a real situation the number of norms to be manipulated can be to too high. The manipulation of norms can be too complex in some cases. Functionalities such as: conflict detection, and versioning could improve the NBIC efficiency.

These requirements are important to the application of the architecture in other services of the Project, which will be used in large scale (e.g. healthcare appointments). However the main objective of the architecture to provide flexibility was achieved in the proof of concept.

6 CONCLUSIONS

E-Gov systems require flexible applications due to the diversity of users and stakeholders. The Semantic and Norm analysis help us to construct a detailed modelling of the organisation and the communication process in a distributed system. In addition it also provides an indication of what are the static and volatile requirements.

Architecture and tools take advantage of this property for the construction of flexible distributed systems. This architecture makes use of SOA principles to provide links between the Norm specification in high level interfaces and the changes at different and heterogeneous components. A case study was conducted in order to evaluate the architecture.

As further work we propose the improvement of the architecture in order to be more flexible, scalable and secure; we are also improving the support for norm specification and manipulation. A deeper analysis of the norm changes occurrences and the capabilities to reflect changes in e-Gov systems will be also investigated in next steps.

ACKNOWLEDGEMENTS

REFERENCES

Bonacin, R., Baranauskas M.C.C., and Liu, K. (2004). Interface Design for the Changing Organisation: an organisational semiotics Approach. In *Virtual, Distributed and Flexible Organisations*, Kecheng Liu (Ed.), Kluwer Academic Publishers, 15, 236-256.

Bonacin, R., Simoni, C. A. C., Melo, A. M., and Baranauskas, M. C. C. (2006). Organisational Semiotics: Guiding a Service-Oriented Architecture for e-Government. Proceedings of *International Conference on Organisational Semiotics*, M. C. C. Baranauskas and K. Liu (Eds.), 47-58.

Fu, Y. and Liu K. (2006). Design of Interoperable Content Management Using Semiotic Methods. Proceedings of *International Conference on Organisational Semiotics*, M. C. C. Baranauskas and K. Liu (Eds.), 37-46.

Husted T., Dumoulin C., Franciscus G., Winterfeldt D., and McClanahan, C. R. (2002). *Struts in Action: Building Web Applications with the Leading Java Framework.* Manning Publications.

Kahler, H, Morch, A., Stiemerling, O. and Wulf, V. (2000). Special Issue on Tailorable Systems and Cooperative Work, *Computer Supported Cooperative Work*, Kluwer Academic Publishers, 1 (9).

Liu K. (2000). *Semiotics in information systems engineering*, Cambridge. University Press.

Marchionini, G., Samet, H., and Brandt, L. (2003). Digital government. *Communications of the ACM*, 46(1), 25-27.

Newcomer E., and Lomow, G. (2004). *Understanding SOA with Web Services*. Addison Wesley Professional.

Peirce, C.S., 1931-1958. *Collected Papers*, Cambridge, Mass: Harvard University Press.

Sprott D. and Wilkes L. (2004). Understanding Service-Oriented Architecture, *CBDI Forum*, Retrieved November, 27, 2006, from msdn.microsoft.com/library/en-us/dnmaj/html/aj1soa.asp

Stamper, R. K., Althaus, K. e Backhouse, J. (1988) MEASUR: Method for Eliciting, Analyzing and Specifying User Requirements. In Olle, T. W., Verrijn-Stuart, A. A. e Bhabuts, L. (eds.) *Computerized Assistance During the Information Systems Life Cycle.* Elsevier Science, Amsterdam, 67-116.

Thatte, S., Andrews, T., Curbera, F., Dholakia, H., Goland, Y. Klein, J., Leymann, F., Liu, K., Roller, D., Smith, Trickovic, I., and Weerawarana, S. (2003). *Business Process Execution Language for Web Services Version 1.1.* Retrieved November, 27, 2006, from http://www-128.ibm.com/developerworks/library/specification/ws-bpel/