



A One-Stop Government Architecture based on the GovML Data Description Language

Olivier Glassey

► To cite this version:

Olivier Glassey. A One-Stop Government Architecture based on the GovML Data Description Language. 2nd European Conference on EGovernment (ECEG 2002), Oct 2002, Oxford, United Kingdom. pp.9. hal-00915141

HAL Id: hal-00915141

<https://hal.science/hal-00915141>

Submitted on 6 Dec 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A One-Stop Government Architecture based on the GovML Data Description Language

Olivier Glassey

Swiss Graduate School of Public Administration (IDHEAP)

Route de la Maladière 21

1022 Chavannes-près-Renens, Switzerland

olivier.glassey@idheap.unil.ch

Abstract In previous research work we defined a conceptual model of a one-stop public administration and we created a prototype to illustrate it. In parallel we participate in an IST research project that is developing an integrated e-government platform and creating a description language based on XML and called GovML. We decided to integrate both works and in this paper we explain how we extended our conceptual model to support GovML and to generate data descriptions for administrative services.

Keywords Public sector, One-stop government, electronic services, modelling, description language, process models

1. Introduction

New technologies are used in many public administrations to deliver improved or new services to their clients. This field of work is commonly called e-government and is becoming increasingly important for administrations and their commercial partners, as well as in the academic world. In previous research work we defined a conceptual model of a one-stop public administration, studying both the structural and behavioural aspects of such a system. Indeed we believe that many researchers and public administrations need stable models of processes that can help them managing the complexity and the rapid evolution of new technologies. In the second section of this paper we explain briefly the steps we went through in order to build a model of a virtual one-stop public administration, from gathering users requirements, both amongst civil servants and citizens, to building a conceptual model. We believe there is a need for conceptual methodologies in that field and various researches were done in that field, such as the GAEL (Guichet Administratif En Ligne) project in Switzerland (Chappelet & Le Grand 2000) or the BTÖV (Bedarf für Telekooperation in der Öffentlichen Verwaltung) method in Germany (Gräslund & al. 1996). Moreover it has to be noted that a Swiss working group on e-government appointed by the federal government has made similar remarks (GSCI 2000). Furthermore this group identifies the definition of patterns for such applications as one of three essential action domains for e-government. In that regard, let us also mention that we developed a prototype of One-stop public administration based on a component architecture distributed over the Internet, in order to validate and to refine the conceptual model.

However the goal of this publication is not to go into the details of the model and of the prototype but rather to explain our current research that aims at expanding this conceptual model with work done in parallel within a European project named “An Integrated Platform for Realising Online One-Stop Government (eGOV)”. This IST research project is lead by a consortium of commercial companies, public administrations and academic partners. Its main goal is to develop an integrated eGovernment platform and we will shortly describe it in section 3. In the fourth section we will present the part of the project that is relevant to this publication: the consortium defined a description language based on XML and inspired by the dialect ebXML. Called GovML this language should enable data flow exchange amongst public administrations and provide a valuable tool to build electronic services. In our own

research work we are currently extending our conceptual model to integrate GovML and to support automatic generation of data descriptions for administrative services. In order to realize that we rely on existing CASE tools and we are developing specific stereotypes to model electronic services. We are also in the process of creating dedicated scripts that will support the generation of GovML descriptions. This work will be explained in section 5.

2. Conceptual Model

2.1 Field study and users requirements

We studied administrative processes in a large public administration in Switzerland, for there was a need to clearly identify and normalize all these different processes (internal processes, relations with its clients, services, procedures, etc.) before it was possible to dematerialise them (Baquias 1999). We spent six months in the "Administration Cantonale Vaudoise" (ACV) and we met project managers, domain managers, departmental managers and users representatives for a total of 28 formal interviews. We also had many casual talks with different civil servants. This field research allowed us to collect a large amount of data on administrative processes, but it was rather heterogeneous since we met various people with very different focuses. We had to find a way to process that data and we will see below how we managed to do it. Moreover we conducted two online surveys (1998 and 2000) in order to discover the expectations of the citizens in the field of electronic administrative services and we had close to 500 questionnaires to analyse.

<i>Service name</i>	Public health inspection
<i>Short service description</i>	The department of Public Health has to inspect restaurants, shops, enterprises, etc. to verify if they comply with public health standards. It makes routine controls where the clients are aware that they will be checked and it also sets up unexpected controls. The results are made public.
<i>Supplier</i>	Department of health
<i>Clients</i>	Shops, restaurants, enterprises, medias, citizens
<i>Uses types</i>	Initiation, termination, notification, validation, publication, modification, consultation, search
<i>Intermediaries/ partners</i>	Consumers association, laboratories

Figure 1: Summary card

Figure 1 shows a summary card of a simplified example of a public health inspection service. After the interview stage, we filled in these summary cards to begin with our iterative modelling process. These cards are loosely based on CRC cards (Class-Responsibilities-Collaborators), an informal class modelling technique with some distinct advantages over simply starting with UML class diagrams (Harmon & Watson 1998). For detailed information on this technique, we recommend (Bellin & al. 1997). These cards helped us collecting the input from the various people we met and transforming it into an object model through an abstraction process. For that model we focused on the services that the ACV delivers to its different types of clients, through different intermediaries or partners.

With the data gathered during our field research we prepared over 90 summary cards, although we did not go into details for each of these administrative services. From these cards it was fairly straightforward to define class descriptions, such as shown in Figure 2. We settled for 65 types of services that an administration can deliver to its clients and classified them according to the widely used typology of Information-Communication-Transaction services. We also defined 50 types of clients that we will present below.

<i>Service class</i>	Public health inspection
<i>Super-classes</i>	Information - Communication
<i>Sub-classes</i>	
<i>Methods</i>	Initiated - Terminated - Notified - Validated - Published - Modified - Consulted - Searched
<i>Attributes</i>	Name - Description - Key_words - Supplier - Clients - Intermediaries - Supports - Payment
<i>Constraints</i>	
<i>Collaborators</i>	InstitutionalCI - BusinessCI - InternalCI

Figure 2: Class description

2.2 Structural model

We worked on this model using object-oriented (OO) technology and the graphical notation language UML (Unified Modelling Language) and we discussed this choice in details in one of our working paper. Here we assume that the reader has good notions of OO technology, but further reference on UML can be found in (Booch & al. 1999). This object-oriented notation language has a very wide base of users, is an Object Management Group standard and combines the strengths of various object methods. It offers nine types of diagrams, amongst them the use cases diagrams that are very useful to formalize users needs. Besides the notation has been extended to model Web applications (Conallen 2000). Such as stated in (Booch & al. 1999), we build models so that we can better understand the system we are developing. Indeed a good model allows us to specify the structure and the behaviour of a system, that is static and a dynamic view of the system. In order to build the structural blueprint we used the class descriptions and the summary cards. We also worked on scenarios and use cases for the creation of the behavioural model, but we will not discuss that here, as it does not integrate with the GovML extension.

The structural model is made of class and object diagrams that show their attributes, their methods and the relationships between them. To build the hierarchy of the class diagrams from the class descriptions and the summary cards, we began with a fairly simple base diagram (Figure 3). A client requests a service from an administration and the supplier of this service can call different intermediaries if necessary, before delivering the service on whatever support suits best.

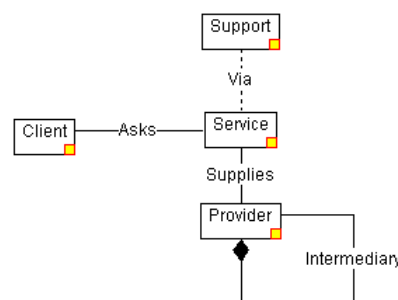


Figure 3: Base diagram

In order to build these class diagrams we used a CASE tool that allowed us to create cascading diagrams. That means that users can navigate through the class hierarchy by a simple click of the mouse. It is out of the scope of this paper to show the operations and the attributes for all the classes, however we designed two synthetic diagrams showing the services and the clients hierarchy (Figures 4 and 5). Furthermore we created an index of over a hundred themes that can be used to classify services.

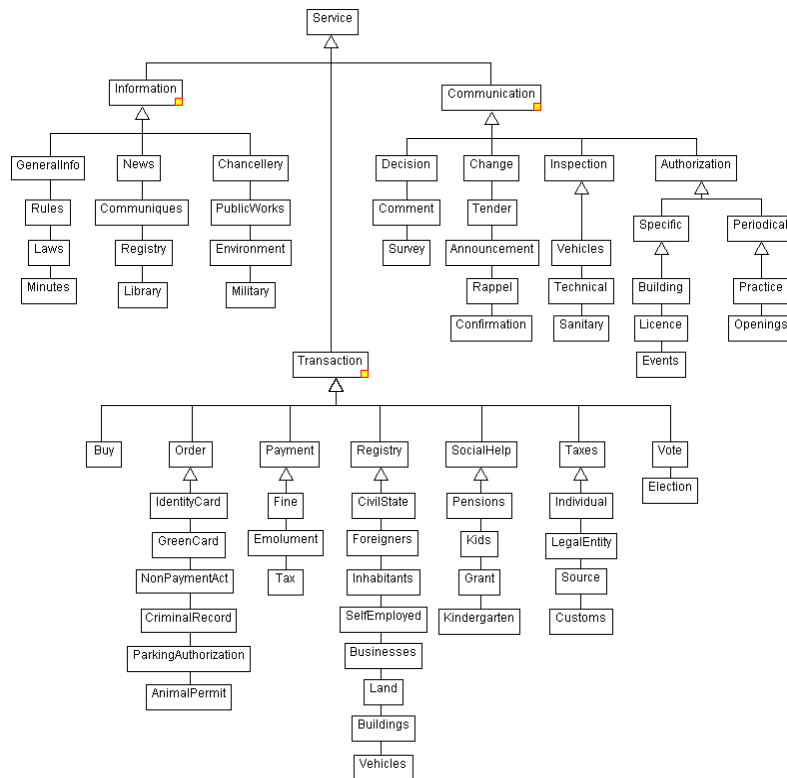


Figure 4: Typology of services

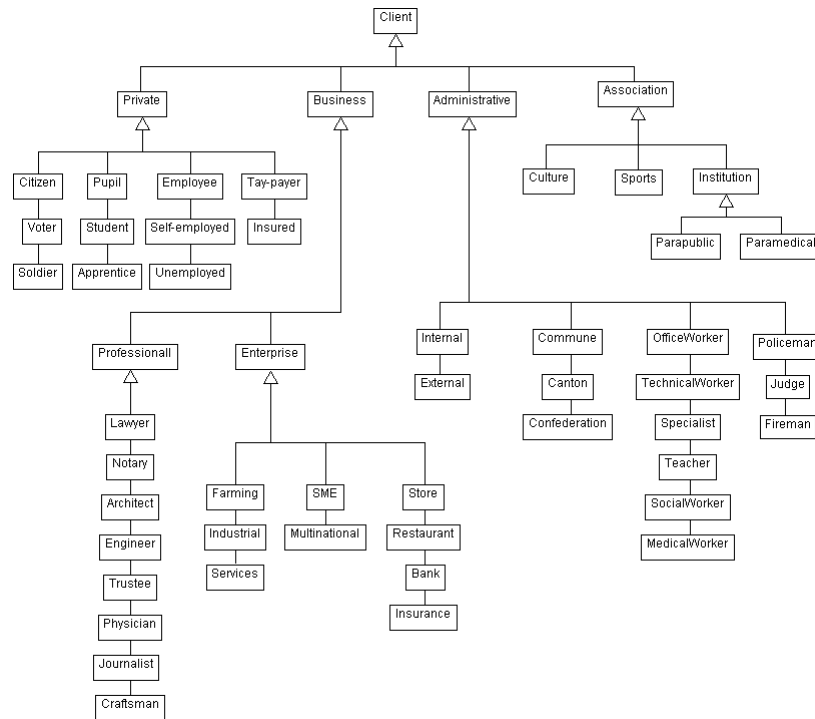


Figure 5: Typology of clients

We also identified the actors of the service delivery: suppliers, clients, intermediaries, etc. Then we established a typology of the suppliers: we decided to divide them into 6 general functions and 28 departments or offices. These divisions do not correspond to a real administration structure, but we think this functional visualization can well be used to represent real administrations with very different organizational structures. Last we defined the ways services can be provided (synchronous, asynchronous) and the supports to be used: mail, fax, electronic delivery, face-to-face, etc. Here we really want to emphasize the fact that this class model represents one point of view of a public administration in Switzerland and might not be of direct use elsewhere. However it can be modified or extended at any point and one does not have to use it as a whole. Although we realize this is only a partial representation of the reality and do not think that this is exhaustive, we believe that it does a good job covering what roles a public administration has to play in a modern society and that it constitutes a good backbone for our model.

3. eGov Project

The eGov platform is being developed in a two-years research project (2001-2003) financially supported by the European Commission under the Information Society Technologies Program of the Fifth RTD framework. The main objective of this project is to develop, deploy, demonstrate and evaluate an integrated platform for realizing online one-stop government. In order to achieve this, the consortium has to:

- Develop a governmental portal that offers advanced features to the citizens: personalisation, multilingualism, authentication, accessibility, etc.
- Develop content and service repositories (both at the national and local levels), as well as a service creation environment used to administrate the repositories.
- Implement a governmental mark-up language (GovML) that supports the exchange of data within the platform.

- Create and supply a number of administrative services to be used during the testing phase.
- Deploy and evaluate the platform in Austria, Greece and Switzerland at the national and local levels.

The general architecture of the platform is illustrated in Figure 6. Citizens and businesses access administrative services through the portal where these services are categorized with “life-events” and “business situations” metaphors. As the portal is linked through Internet and GovML to the national service repository and the local service repositories, users can obtain services from different administrations at various levels in a transparent and integrated manner. If they request information services they will get the data from the relevant repository, and if they need transaction services, these will be executed in the service runtime environment before the results are send back to the users through the portal. On their side, the public administrations maintain their service repositories with the service creation environment.

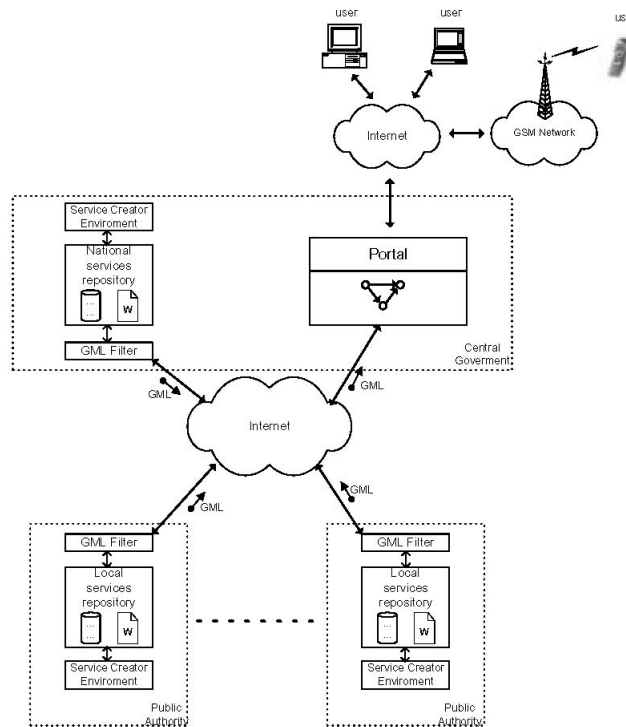


Figure 6: Overview of eGov platform

Details on one-stop government and on the eGov project are available at <http://www.egov-project.org/>, for example in (Tambouris 2001) and (Wimmer & Traunmueller 2002).

4. GovML Description Language

As mentioned above, one of the main goals of the eGov project is to define a description language that supports the exchange of data within the platform. In order to do so, the consortium conducted a survey of international best practices, both in the domain of XML and of governmental data exchange frameworks. It then identified the requirements of the project and the data elements necessary to create the GovML sub-vocabularies. With the input of all partners and after several iterations to refine the data models, the final vocabulary

was specified according to the relevant ISO/IEC standard and the XML schemas were developed.

Two categories were identified for content description. Common content for all public authorities that is created only once for all and that is centrally stored at the governmental content repository is called **Content Data**. Specialized content related to service provision and execution by a specific public authority is identified as **Service Data**. GovML vocabularies for Content and Service Data are inspired by ebXML (<http://www.ebxml.org/>). Moreover GovML vocabulary for **Metadata** elements related to information resources was created. It is based on RDF schemes (<http://www.w3.org/TR/rdf-schema/>) and compliant with the Dublin Core Metadata Initiative (<http://dublincore.org/>) and with the electronic Government Metadata Standard (eGMS), a part of the British e-Government Interoperability Framework (<http://www.govtalk.gov.uk/>).

Complete specifications of the GovML vocabularies are out of the scope of this paper and they are available on the project website. However we mention some GovML features in order to show the general ideas behind this language. For the **Content Data**, 19 attributes were selected, such as *Title*, *Description*, *Language*, *Related Services*, *Audience*, *Periodicity*, *Time to deliver*, *Cost Information*, etc. They are used to describe an administrative service in a general way. For precise information on a specific service, the **Service Data** vocabulary adds attributes such as *Public Authority Name* and *Department*, *Eligibility*, *Contact Details*, *Procedure to follow*, *State of a service*, *Delivery Channel*, etc. **Metadata** will be used in the eGOV platform to search and retrieve governmental resources across the Internet and to develop the navigation tools of the portal based on life-events and business situations. Twelve metadata attributes were selected, such as *Title*, *Creator*, *Publisher*, *Format*, *Language*, *Has Translation*, etc.

The eGov platform will provide a tool that allows administrative users to create GovML content and documents with a GovML syntax that will conform XML schemas. These documents can be translated in different languages and will be used respectively in service repositories (content and service Data) and in the portal (metadata). We believe that the GovML description language and the supporting architecture of the platform is one of the major strengths of the eGov project and therefore we want to integrate it in our own work. In the next section we will show how we propose to use it with the one-stop government model we developed previously.

5. Proposed One-Stop Government Architecture

In (Glassey 2002) we explained how we built a prototype of one-stop government based on a conceptual model and on use cases: we developed UML class models with a CASE tool and we used them to generate code for the backbone architecture of the system. Using a similar approach, we extended this model to be able to generate GovML files. We added all the GovML data attributes from the different sub-vocabularies to the relevant class models in order to provide the users with a tool that supports them in the GovML content creation process. We also want to adapt or develop GovML filters that will ensure the automatic generation of the GovML description files from the conceptual model, but this work is not complete yet.

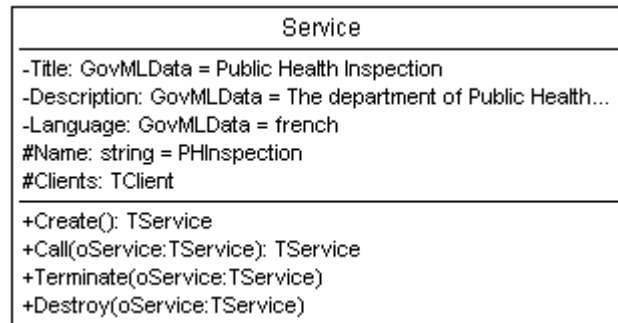


Figure 7: Example of Service Class

Figure 7 shows a simplified example of a Service Class: it contains some of the attributes and the methods that we used for the code generation and it adds specific GovML attributes. The users are provided with a simple graphical interface to enter GovML content and the CASE tool can then automatically validate this content, ensuring the integrity of the data model and its conformity to the XML schemas. Furthermore it takes advantage of the inheritance properties of UML class model to distribute these attributes in the sub-models and the sub-classes. We believe that this conceptual architecture offers some distinctive advantages. It supports the integration of the technical parts of the system (such as the object models and the code generation) with the data description parts. Indeed these different tasks are realized by different workers and teams that have very different problems to cope with: it gives them a common “language” and a graphical model to achieve the integrated development of a one-stop government system. It also supports the integration of heterogeneous systems and administrative services with a standardized description language that allows the exchange of data between them. Finally it can also be used to verify the proper description of administrative services and the appropriate creation of metadata.

6. Conclusions

Although we did not present the one-stop government architecture we developed and the GovML language in full details, we think that we showed the general concepts of this architecture, as well as its possible applications. As this is still a research in progress, we intend on validating it within our work in the eGov project and it is very likely that we will yet modify and refine this model. We believe that it is of importance to develop such data models and supporting tools to help public administrations creating and delivering electronic administrative services. Indeed we think that the GovML language will allow the exchange of data between different components of a one-stop government system and that this is a necessary condition to provide true electronic services. However we realize that this architecture stays rather theoretical and cannot be used as is in public administrations: it is rather to be taken as a general reference for e-government projects.

7. References

Baquiast, J.-P. (1999) “Le guichet unique dans l’administration française”, *New interfaces between Administration and Citizens, One-stop-government through ICT*, International Workshop, Bremen, Germany.

Bellin, D., Suchman Simone, S. & Booch, G. (1997) *The CRC Card Book*, Addison-Wesley, MA.

Booch, G., Rumbaugh, J. & Jacobson, I. (1999) *The Unified Modelling Language User Guide*, Addison-Wesley, MA.

Chappelet, J.-L. & Le Grand, A. (2000) "Towards a Method to Design Web Sites for Public Administrations", *IFIP Working Conference on Advances in Electronic Government*, Zaragoza, Spain.

Conallen, J. (2000) *Building Web Applications with UML*. Addison-Wesley, MA.

Glassey, O. (2002) "A One-Stop Government prototype based on use cases and scenarios", *eGovernment: State of the Art and Perspectives*, *eGov Conference within DEXA 2002*, Aix-en-Provence, France.

Gräslund, K., Krcmar, H. & Schwabe, G. (1996) *The BTÖEV Method for Needs-driven Design and Implementation of Telecooperation Systems in Public Administrations*, Universität Hohenheim, Stuttgart, Germany.

GSCI (2000) *Guichet virtuel: la communication électronique avec l'administration, le Parlement et les tribunaux*, Groupe de Coordination Société de l'Information, Switzerland.

Harmon, P. & Watson, M. (1998) *Understanding UML: The developer's guide with a Web-based application in Java*. Morgan Kaufmann, CA.

Tambouris, E. (2001) "An Integrated platform for Realising Online One-Stop Government: The eGov Project", *Proceedings of the DEXA International Workshop "On the Way to Electronic Government"*, IEEE Computer Society Press, CA.

Wimmer, M. & Traunmueller, R. (2002) "Towards an Integrated Platform for Online One-Stop Government", *ERCIM News*, Special Theme: e-Government, Issue 48.