A One-Stop Government Prototype Based on Use Cases and Scenarios

Olivier Glassey

A One-Stop Government prototype based on use cases and scenarios.

Olivier Glassey
Swiss Graduate School of Public Administration (IDHEAP)
And INFORGE - University of Lausanne
Switzerland
olivier.glassey@idheap.unil.ch

1. INTRODUCTION

In previous work we defined a conceptual model of a One-Stop Public Administration, putting the accent on both the structural and behavioural aspects of such a system. We had decided to work on such a model because we believe that there is a need for conceptual methodologies in that field: various researches are being conducted, 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]. Indeed many researchers and public administrations need a stable model of processes that can help public administrations managing the complexity and the rapid evolution of new technologies. This is also what a Swiss working group on e-government appointed by the federal government thinks [GSCI, 2000]. Furthermore this group identifies the definition of patterns for such applications as one of three essential action domains for e-government.

During our modelling work, we took into account the internal processes of an administration, its relations with the customers and the expectations of the citizens: we made a six month survey in the "Administration Cantonale Vaudoise" (ACV), a large public administration at the cantonal level in Switzerland. There 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. 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 analyze. We also took into account the different targets that a One-Stop public administration can have (citizens, businesses, civil servants, other administrations, etc.) and the different distribution canals (Internet, wireless, public kiosks, administration's Intranet and so on). Using that field data we filled in "summary cards", inspired by CRC cards (Class-Responsibilities-Collaborators). For more information on this technique, we recommend [Bellin & al., 1997]. First these cards helped us collecting the input from the various people we met and transforming it into a structural model through an iterative abstraction process. We defined this class model using the Unified Modeling Language (UML), developed by [Booch & al., 1999]. Then we used the cards to create use cases and scenarios following the methodology developed by [Kulak & Guiney, 2000]. We will not go into the details of this model because it is not the aim this paper where we want to explain how we built a prototype based on this model. In order to validate and to refine the conceptual model we decided to build a small system of a one-stop public administration and throughout this paper we will explain the different steps of this work. It has to be noted that it is only small prototype will limited functionalities, because we lacked the resources to develop a full-scale system. However we think that the interest of this paper is focused on the methodology we used to build the prototype rather that on the final system. That is also why we will not show any implementation details and we will stay at the general software architecture level.

2. USE CASES AND SCENARIOS

We wanted to illustrate rather typical *Information-Communication-Transaction* administrative services and we selected ten functionalities that we wanted to be part of the prototype. For each one of these we built use cases, scenarios and sequence diagrams. Here we shortly explained why we chose these particular functionalities even though they may seem quite simple :

- **Content publication** : this functionality is typical of *Information* services and it demonstrates problems of content management, validation, update and access rights, as well as the ability of non-technical persons to publish content without specific knowledge.
- **Directory** : this service is at the border of *Information* and *Communication* services as the clients can get in contact with the administration from within the directory by email, but they can also phone, fax or send postal mail; we also found during our surveys of the citizens that it was one of the most expected service.
- **Calendar** : this tool illustrates typical problems of dynamic content (*Information*) management and interfaces with databases; it is also at the interface with the following point, as a client can add public events to his personal agenda and, by doing so, receive a reminder.
- **Appointments and personal agenda** : the ability for a user to have his own agenda is a limited personalization capability in the prototype; the fact that a client can propose an appointment to a civil servant is the beginning of a very simple workflow (or very limited *Transaction* services).
- **Validation and acceptation of an appointment** : this shows limited workflow functionalities and interactions with a human actor, for the system checks for conflicts in appointments and a person in charge validates it before the user receives a confirmation.
- **Discussion forums** : this *Communication* service is an interesting feature because many forums are third party software and sometimes remotely hosted, so it allowed us to show the integration of commercial components or application in a One-Stop public administration.
- **Chat** : basically we chose to add a chat to the prototype for the same reasons as above, with the idea of a future integration with the agenda, allowing the client to make an appointment for a official chat with a civil servant, which would provide advance *Communication* services.
- **Search engine** : this is a very useful feature to find one's way in the mass of information provided by online government and it shows the ability to integrate distant functionalities within our prototype; a search engine is quite simple, but we can imagine to add a payment or a certification "engine" in a real system, which are necessary tools to support complex *Transaction* services.
- **Official forms** : official forms are the bases of a public administration's operations and they are at the convergence of *Information*, *Communication* and *Transaction* services; they are also the most requested feature that came out of our surveys.
- **Tax declaration** : this is the only really *Transaction* service of the prototype and we added it because it helped us illustrate the integration of legacy systems (we developed a prototype of tax declaration in 1997).

For each of these ten functionalities we created use case diagrams such as shown in Fig. 1 and we developed basic scenarios [Fig. 2], but here we will only show them for the first feature of the prototype. The use case diagrams and the scenarios are very useful to have a common language between technical persons, users, managers and leaders. They also provide a solid basis for a posteriori tests and validation of a system.

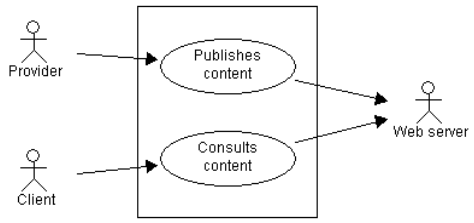

Figure 1 : Use case diagram

Use case name	**Content publication**
Abstract	The public worker (or service provider in our terminology) publishes content on the Web: news, job offers, etc. The client can consult this content online after it as been validated by a person in charge.
Normal sequence of events	1. The provider publishes new content 2. The content is validated 3. The content is made available online 4. The client consults the content he is interested in.
Alternate sequence	4. The client does not consult any content
Exception handling	At step 2, the content does not go online if it is not validated. A specific procedure begins.
Triggers	
Assumptions	
Preconditions	New content has to be published.
Postconditions	Content is accessible on line.
Authors and date	Olivier Glassey, February 12th, 2002

Figure 2 : Basic scenario

In order to complete the scenarios, which show in what order a procedure is accomplished, we defined business rules [Fig. 3] to explain how a particular activity is conducted, using the methodology developed by [Ross, 1997].

Business rule	**Job offer publication**
Short description	All job positions to be filled must be made publicly available, event if they are filled internally later on.
Source	Job regulation in public administrations
Type	Fact

Figure 3 : Business rule

When the scenarios are a textual representation of a procedure or a service, UML provides a graphical way of describing it. Indeed the sequence diagrams [Fig. 4] show the different steps of a procedure, but they also add information by introducing classes, actors and messages. They provide a solid basis for developers who can later on work on activity and state diagrams before they begin to actually work on the code. Furthermore there are many CASE tools that do code generation and reverse engineering, thus reducing greatly the workload of the software engineers.

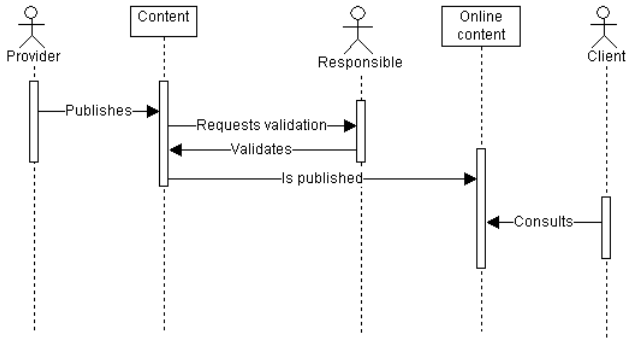

Figure 4 : Sequence diagram

As we mentioned above, we created such diagrams for each of the ten functionality of the prototype, then we used our structural model in conjunction with a CASE tool to create the class model of our prototype. This allowed us to generate its software "backbone", but we also needed to define logical software architecture before we started to really implement this system.

3. ARCHITECTURE OF THE PROTOTYPE

We needed a software architecture that allowed us to take into account the following constraints, that we had found were very strong in public administrations :

- Ability to access heterogeneous applications and platforms
- Simple user interface and openness to the Internet
- Use of standard and robust technologies
- Modularity et scalability

To create the middleware layer we chose a distributed component architecture because it fitted our needs best and because there are really strong standards on the market. We will not present the characteristics of this architecture here, but we recommend [Fingar & Stikeleather, 1996] which is one of the foundation papers of this model. The three main components communication models are CORBA from the Object Management Group, COM+ (and now .NET) from Microsoft and Java/RMI for Sun Microsystems. Neither will we discuss these here, as there is an excellent comparison in [Suresh Raj, 1998]. Let us just say that for reasons of resources and simplicity we built our prototype using the COM+ model for the back-office applications. To be able to distribute these services to the clients we decided to build a dynamic Web interface based on the PHP/MySQL couple (respectively a server script language and a database, both of which are Open Source). The output displayed in the end-user interface is pure HTML, which means that it can be read by any browser such as Internet Explorer, Netscape or Opera.

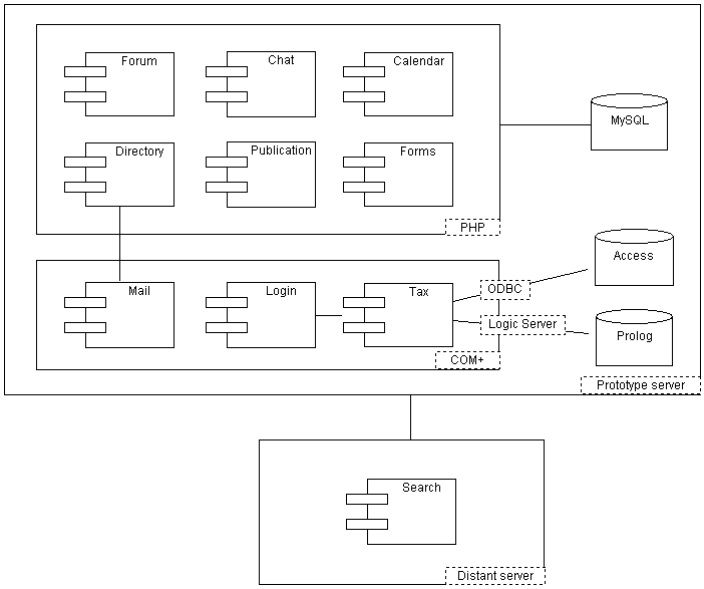

Figure 5 : Architecture of the prototype

Figure 5 shows the general logical architecture of the prototype and of its functionalities. For most of the system, we adapted existing PHP applications (which can be seen also as components) : forum, chat, calendar, directory, content publication tools and access to the official forms. On the other hand, the tax declaration application we built in 1997 was written in Delphi and used the ODBC gateway to connect to a Microsoft Access database and a Logic Server to retrieve calculation rules in Prolog. We simply wrote a COM component that encapsulates this application and is able to be distributed over the Internet using the ASP scripting language. We also added a commercial component called ASLogin that allowed us the secure the tax declaration pages. Finally we added the Google search engine in our prototype. We will not go into anymore details regarding the implementation of this prototype as this is rather a "toy" compared to real systems and because we think the interesting concept here is the general architecture : it allowed us to integrate various commercial and Open Source components and applications into what appears to be a unified and transparent system with a single interface.

4. OPERATION OF THE PROTOTYPE

The prototype can be tested at *http://uts.unil.ch/vade/*, but it is only available in French. Here we will briefly describe its operation and show how users with different access rights can accomplish various tasks within a single interface. Let us mention that we made no particular work on the ergonomics of the prototype as it was not our intent. We used very classical navigation techniques with a left menu based on the *Information-Communication-Transaction* typology and an upper menu that shows the hierarchy of the pages. We realise that this is not entirely satisfying and we will work on a different interface in the future, most likely based on the "life events" metaphor.

The first screen [Fig. 6] shows the interface that a regular client (with no particular rights) sees when he chooses to see the latest news. The second one [Fig. 7] shows how a user with specific access right can publish new content within the same interface. Without any HTML knowledge he or she can publish text, links or images that will be validated before going online. Another example (not shown here) of clients with different access rights using a similar interface is the directory : a regular client can only browse or search the directory when selected civil servants have the access to add, edit or delete entries, in that case with additional links in the interface.

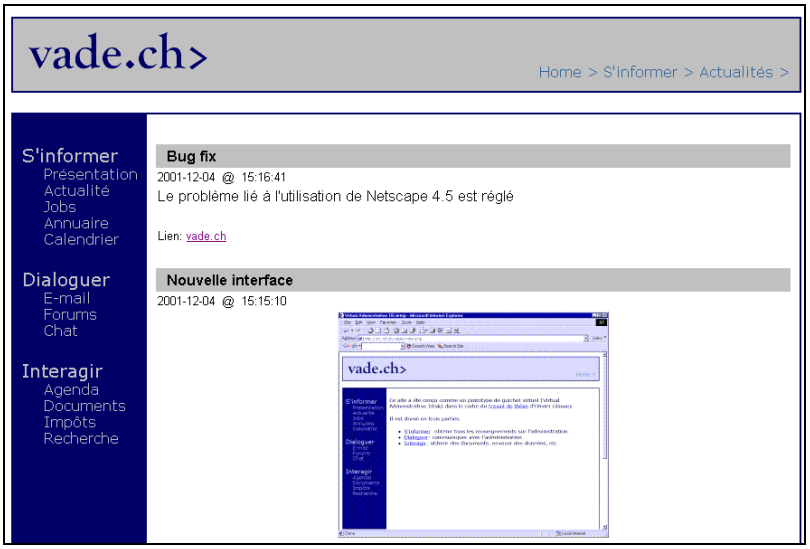

Figure 6 : Interface to access content

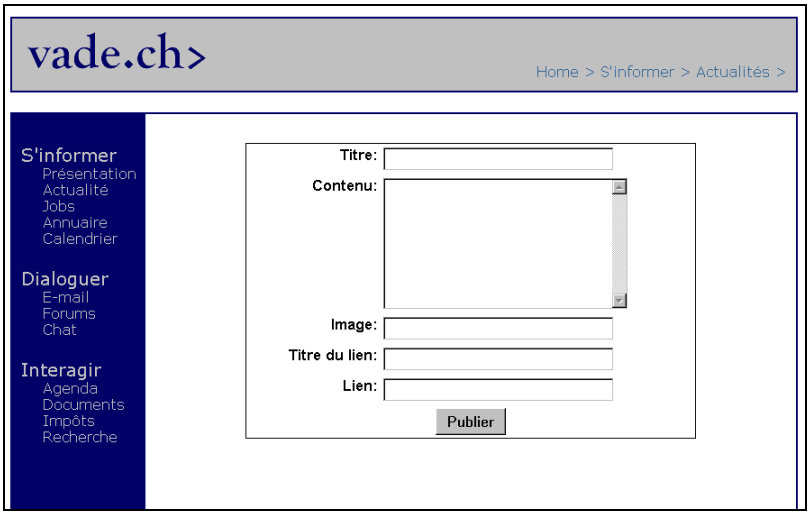

Figure 7 : Interface to publish content

The prototype in itself does not show anything very impressive, but we think that its strong point is the concept of a single and universal interface that can be used via any distribution channel and by any type of client.

5. CONCLUSIONS

What we have shown in this paper is the application of a conceptual model and methodology to a practical case of One-Stop Government. The result of this work is a simple but functional prototype. We now want to emphasize what we believe are the strong points of this work. First we think that it is almost necessary to find a common language between technical persons, users and managers, and we found that the use of graphical models developed in UML provide a good way to achieve that. We also found that scenarios and sequence diagrams, although they are simple enough to be understood by anyone, are a great basis for software engineers to conceptualise and implement information systems that satisfy the needs of their users. We also realised that we needed a solid and standard logical architecture in order to integrate heterogeneous applications and software components into a seemingly unique One-Stop Government system. Furthermore we think that, in order for this system to be accessible from anywhere and by anyone via a standard user interface, the best choice was to use a middleware layer interfaced with the Internet. In the future it is also very likely that this middleware will have to be interfaced with wireless technologies. Thus the architecture we chose allowed us to build a dynamic and universal interface that constitutes a single entry point to very heterogeneous electronic administrative services, providing true One-Stop Government capabilities.

6. BIBLIOGRAPHIE

[Bellin & al., 1997]
Bellin, D., Suchman Simone, S. & Booch, G. *The CRC Card Book*. Addison-Wesley, Massachusetts, USA, 1997.

[Booch & al., 1999]
Booch, G., Rumbaugh, J. & Jacobson, I. *The Unified Modeling Language User Guide.* Addison-Wesley, Massachusetts, USA, 1999.

[Chappelet & Le Grand, 2000]
Chappelet, J.-L. & Le Grand, A. Towards a Method to Design Web Sites for Public Administrations. In : Galindo, F. & Quirchmayr, G. (Eds.). *Advances in Electronic Government.* IFIP WG 8.5 in Zaragoza, Spain, 10-11 February 2000.

[Fingar & Stikeleather, 1996]
Fingar, P. & Stikeleather, J. Distributed objects for Business. Sunworld, USA, 1996.

[GCSI, 2000a]
GCSI. 2e rapport du Groupe de Coordination Société de l'Information à l'intention du Conseil Fédéral. Suisse, 2000.

[Kulak & Guiney, 2000]
Kulak, D. & Guiney, E. *Use Cases : Requirements in Contexts.* Addison-Wesley, Reading, USA, 2000.

[Ross, 1997]
Ross, R. *The Business Rule Book : Classifying, Defining and Modeling Rules, Version 4.0*. Business Rules Solutions, Inc., USA, 1997.

[Suresh Raj, 1998]
Suresh Raj, G. A Detailed Comparison of CORBA, DCOM and Java/RMI. Wisconsin, USA, 1998.