

Federation of E-government: A Model and Framework

By *Abdelbaset Rabaiah and Eddy Vandijck*

Local governments, including municipalities in modern democracies, have an elevated level of autonomy. This stems from one of the golden rules of a democratic society: division of authority. Traditionally, priorities and budgeting in light of national and domestic laws direct the work of local governments. This differs significantly from businesses, where decisions are centrally made.

In democracies, local governments have semiautonomous status. Resistance to change on the side of government employees can be more evident and severe than in businesses. There is also a tendency to keep information private within departments of local governments. This renders much of the integration efforts ineffective. Rebuilding of the IT infrastructure entails replacement of hardware and software in addition to training employees on the new systems. The costs can be far above the ground. A staggering 35 percent of e-government projects are total failures, 50 percent are partial failures, and only 15 percent are success stories.¹

That said, it could be better and cheaper to follow a different approach. This article introduces the Federation of E-government model. A detailed implementation framework of the model will be presented.

Can Business Models Be Applied to Governments?

As the purpose of establishing a local government is different from that of a firm, the nature of each is different and the models to be used for operation of each also should be different. **Figure 1** summarizes the major differences between business and government.

Criteria	Business	Government
Aim	Maximized profit (clearly defined)	Well-being of society (stability)
Driver	Business value	Constituents' needs
Processes	Administrative/technical	Legal/political
Client's relationship paradigm	Consumership	Citizenship
Operation	Centralized	Decentralized/autonomous
Emphasis	Efficiency, profitability	Correctness, credibility

The figure shows that e-government and e-business differ in many ways and, therefore, cannot be approached the same way. Borrowing successful models from business does not necessarily mean that they will also work well for governments. In other words, even if platform integration has worked well for

e-business, it might not work well for e-government. This is where the Federation of E-government model fits in.

The Federation of E-government Model

This model perceives government units as a collection of autonomous entities where there is little or no integration among them. This way, the central government can save a lot of effort and costs in trying to integrate disparate (legacy) systems of the different local governments. With this model, there is no real need to replace incompatible systems. Furthermore, the central government should not pay attention to the way a particular system within a local government works, but should rather be concerned with inputs and outputs of that system. The model is analogous to the well-known object-oriented design methodology. It views a local government or municipality as sort of an abstract "object." Each object is encapsulated, which represents the reluctance to share information as well as the isolation of legacy systems. This definition on the technical level can better fit the political level. The Federation of E-government model, as such, can preserve the autonomous status of local governments and the long-inherited bureaucracy.

The model is not deeply concerned with the detailed workflow or processes within each object (a local government's IT system), but instead is concerned with the "interfacing" among objects (local governments). This renders the whole system platform modeling simpler, cheaper and easier to implement.

Because the whole platform is now partitioned into simpler objects, a government can implement its IT platform incrementally instead of all at once, thereby minimizing the risk of failure.

The model protects autonomy of local governments because local information and processes are kept and preserved locally. Since local information and flow of processes are not accessible from outside the local government, the risk of hacking attempts is reduced, which makes the whole system more secure.

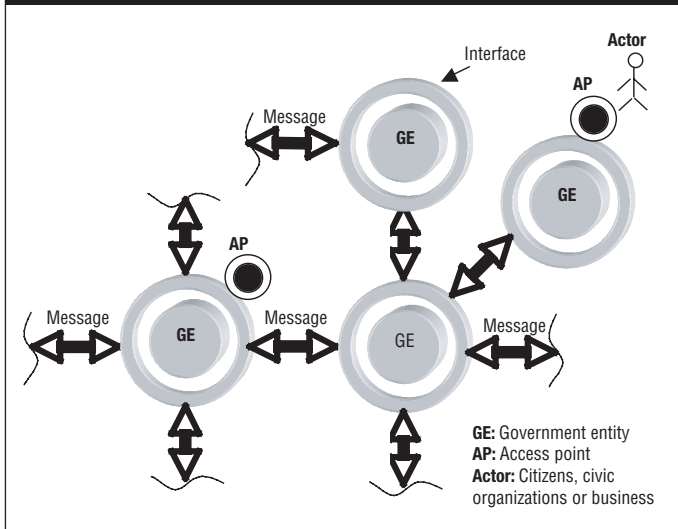
Components of the Model

For the model to be coherent and achieve the objectives set forth previously, it should include some basic components that enable the model to possess the necessary encapsulation and abstraction features mentioned earlier. The following subsections describe these components.

Governmental Entities

Government entities are viewed as black box entities. Each government entity represents an autonomous governmental unit or a municipality. As seen in **figure 2**, each government

Figure 2—Relationships That Exist Among Government Entities Through Interfaces



entity has the ability to interact by sending and receiving messages with one or more different entities within the model through interfaces and messages. A particular government entity does not have to know, or even follow, the same operational procedures as any other government entity. This maintains its assumed autonomy, and legacy systems can remain in place. There is a need, however, to convert format from one legacy system to another. Conversion is done through interfaces.

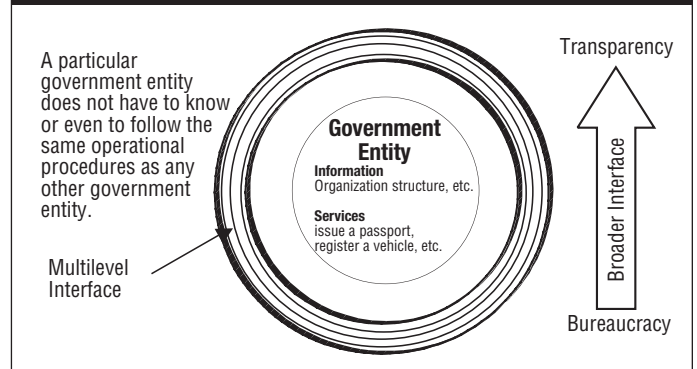
Interfaces

Interfaces achieve the desired separation of responsibilities and distribution of authorities among government entities. They reflect the protocols set forth by the government or the political system. These protocols should be implemented carefully and accurately as they control and coordinate the relationships among government entities, as per regulations set forth by the government. An interface is the security forte of a local government. Built-in firewalls and filters screen incoming and outgoing traffic. Interfaces describe in great detail the type of information that a particular entity can provide (e.g., personal records of a citizen) as well as the set of services that this particular government entity can provide (e.g., issuing a passport). **Figure 3** depicts a typical government entity encapsulated by its interface. Any government entity that needs to interact with another must do so only via its interface. This helps keep the internal operation and data of a particular local government private. Only information set to be public can appear in the interface. The degree of openness of an interface affects the transparency of that local government, i.e., the more open the interface is, the better transparency becomes (see **figure 3**). Services offered by a particular government entity (e.g., a ministry) can vary greatly. Therefore, each interface has multiple levels of interaction.

Messages

As shown in **figure 2**, to request a service, a message must be sent to the concerned local government. These messages can originate at some government entity or at an access point (AP). Messages must have a standardized format. A message

Figure 3—A Detailed View of a Typical Government Entity



can be used to ask for a service or it can be loaded with a whole document. Proper authorizations and identifying information of the sender should be incorporated in each message.

Access Points

APs act as inputs and outputs between the system and the actors (social bodies, including citizens, businesses and civic organizations). An AP can be a traditional help desk, a web portal or any other channel. **Figure 2** shows that there may be more than one AP in the model. Through these APs or "portals," social bodies can request services or information. These requests trigger consecutive messages and activities (processes) throughout the concerned government entity(ies) until the requested service is complete and feedback is delivered through either the same AP or another one.

A Technical Implementation of the Model

A model that is not able to be implemented is hardly useful. This section briefly describes the IT platform architecture that can underlie the model.

Each governmental entity may have its own back-office systems. These could be legacy systems that are very difficult to integrate into other (normally new) systems. Inarguably, this means that different government entities can have different data formats. This should not raise a problem with the model because it was originally built around the idea of encapsulation of disparate systems. However, these systems should be able to "talk" to one another regardless of the hardware and software used in each of the government entities. The idea here is to make common interfaces among the government entities, not to make a common platform. But how can interfaces become common among disparate systems? A lot of research suggests the adoption of a widely supported data format: the Extensible Markup Language (XML). The power of XML over the more traditional HTML format is that it describes not only the content and formatting but also the meaning and properties of each element (object) in the document. A government can come up with its own ontology to define concepts, terminologies, meaning, etc. The advantage is quite obvious when searching through documents. Search results will be conceptual rather than simply matching texts. For multilingual countries, this could be quite advantageous because the search will be concerned with concepts regardless of the original language used to write the document.

In this context, the World Wide Web Consortium (W3C) has developed XQuery based on XML. XQuery can produce more useful results than traditional queries. Searching for specific content can be based on concepts instead of occurrences of the search string and independent of the language used for search. This should yield accurate and structured results. This feature is not only indispensable for multilingual countries, as mentioned before, but it is also vital for people who need to access governmental information or services where the official language is not their mother tongue.

There might be a need, though, for middleware—software used to convert data formats from one form into another—in case the installed software in one of the government entities does not support XML format.

The Federation of E-government model is indispensable when building a large-scale governmental portal. A governmental portal provides one-stop access to governmental services and information. One-stop government refers to the integration of public services from a citizen's point of view.³ A governmental portal can collect information from different government entities simply by sending messages to the interfaces of the government entities concerned. The local content is compiled by the local government entity(ies). This makes it easy to manage complexities of hyperlinks and document formats. Hyperlinks and local URLs are maintained by the government entities. Document formatting, layout and style can then be standardized by applying standard governmental cascading style sheets (CSS) to the received local content. A similar example of such an implementation, which proves workability of the Federation of E-government model, is the Swiss governmental portal (www.ch.ch).

Messages in the Federation of E-government model can be easily implemented using the Simple Object Access Protocol (SOAP). This technology facilitates communication among applications running on different operating systems and different platforms, and developed with different programming languages. There is a new trend among big enterprises to use the service-oriented architecture (SOA). The Organization for the Advancement of Structured Information Standards (OASIS) defines SOA as “a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains.”

Basically, SOA is built around interoperability. The technology is built around the XML document format, so XML provides a robust content description and structuring language.

SOA can be a perfect implementation architecture for the model with some adjustments for e-government. This is a software architecture that best fits loosely coupled systems. SOA facilitates interconnecting legacy systems.

Conclusion

The proposed model simplifies implementation of e-government initiatives. It facilitates back-office and front-office bridging. As this article shows, the model has the capacity to cut down huge expenses; it does not inflict replacement of legacy systems, and training costs are avoided as a result. In addition, it assures minimal changes in the working environment of government employees, as they can still work on their familiar systems. This dramatically reduces the risk of project failure.

The model supports encapsulation or information hiding. Thus, a local government can decide on how transparent it will be. It can expose its internal processes and information more and, thus, improve transparency. If it is a fierce bureaucrat, on the other hand, the local government can hide its internal processes and information. The model can be a double-edged sword. On one hand, the model assimilates bureaucracy but only if the local government is keen on it. In this case, it constricts its interface (see **figure 3**). On the other hand, the model welcomes the opening up of internal processes to enhance transparency. Of course, everybody wants more transparency, but, unfortunately, it is not always promoted by some government entities. The purpose for the model was not to create one that imposes transparency on the technical level and then fails because of the expected conflict with bureaucracy. Instead, it should be made valid for all the government entities, regardless of the level of transparency they espouse.

A downside of the model, though, is that it may hamper innovation. As long as software and hardware are running properly, they will not be changed as often (when an interesting new technology debuts, for example).

References

- Anthopoulos, Leo; Ioannis Tsoukalas; “A Cross Border Collaboration Environment, as a Means for Offering Online Public Services and for Evaluating the Performance of Public Executives,” IEEE Computer Society, USA, 2005
- Bochicchio, Mario; Antonella Longo; “Conceptual Modeling of Data Intensive and Information Intensive Web Applications,” IEEE Computer Society, USA, 2004
- Klischewski, Ralf; Martti Jeenicke; “Semantic Web Technologies for Information Management, Within e-Government Services,” IEEE Computer Society, USA, 2004
- Laudon, Kenneth C.; Jane P. Laudon; *Management Information Systems, 9th Edition*, 2006
- Mugellini, Elena; Omar Abou Khaled; Maria Chiara Pettenati; Pierre Kuonen; “eGovSM Metadata Model: Towards a Flexible, Interoperable and Scalable eGovernment Service, Marketplace,” IEEE Computer Society, USA, 2005
- Rabaiah, Abdelbaset; Eddy Vandijck; Farouk Musa; *Abstraction of eGovernment*, Proceedings of the IADIS International Conference E-Commerce, Barcelona, Spain, 2006, p. 27-34
- Scholl, Hans J.; “Interoperability in e-Government: More Than Just Smart Middleware,” IEEE Computer Society, USA, 2005
- Spahni, Dieter; “Managing Access to Distributed Resources,” IEEE Computer Society, USA, 2004
- Themistocleous, Marinos; Zahir Irani; “Developing E-Government Integrated Infrastructures: A Case Study,” IEEE Computer Society, USA, 2005
- Traunmüller, Roland; Maria A. Wimmer; “Web Semantics in e-Government: A Tour d’Horizon on Essential Features,” Institute of Applied Computer Science, University of Linz, IEEE Computer Society, USA, 2002

Endnotes

- ¹ Punia, Devendra K.; K.B.C. Saxena; *Managing Inter-organisational Workflows in eGovernment Services*, ACM Press, USA, 2004
- ² Grönlund, Åke; *What's in a Field—Exploring the E-government Domain*, IEEE Computer Society, USA, 2005
- ³ Tambouris, Efthimios; “An Integrated Platform for Realising Online One-Stop Government: The eGOV Project,” IEEE Computer Society, USA, 2001

Abdelbaset Rabaiah

is a senior researcher and practitioner in electronic government. He worked on a number of projects within the context of e-government, and has many years of experience in business information management. He can be reached at Abdelbaset.Rabaiah@vub.ac.be.

Eddy Vandijck

is a professor at Vrije Universiteit Brussel, where he teaches information systems, databases and ICT auditing. He can be reached at eddy.vandijck@vub.ac.be.

Information Systems Control Journal is published by ISACA. Membership in the association, a voluntary organization serving IT governance professionals, entitles one to receive an annual subscription to the *Information Systems Control Journal*.

Opinions expressed in the *Information Systems Control Journal* represent the views of the authors and advertisers. They may differ from policies and official statements of ISACA and/or the IT Governance Institute® and their committees, and from opinions endorsed by authors' employers, or the editors of this *Journal*. *Information Systems Control Journal* does not attest to the originality of authors' content.

© 2007 Information Systems Audit and Control Association. All rights reserved.

Instructors are permitted to photocopy isolated articles for noncommercial classroom use without fee. For other copying, reprint or republication, permission must be obtained in writing from the association. Where necessary, permission is granted by the copyright owners for those registered with the Copyright Clearance Center (CCC), 27 Congress St., Salem, Mass. 01970, to photocopy articles owned by ISACA, for a flat fee of US \$2.50 per article plus 25¢ per page. Send payment to the CCC stating the ISSN (1526-7407), date, volume, and first and last page number of each article. Copying for other than personal use or internal reference, or of articles or columns not owned by the association without express permission of the association or the copyright owner is expressly prohibited.

www.isaca.org