Applying the ISO RM-ODP Standard in E-government

B. Meneklis[1], A. Kaliontzoglou[2,3], D. Polemi[1], C. Douligeris[1]

[1] University of Piraeus, Department of Informatics, Karaoli & Dimitriou 80 Str.,
185 34 Piraeus, Greece
{bmenekl, dpolemi, cdoulig}@unipi.gr
[2] National Technical University of Athens, School of Electrical & Computer Engineering,
Heroon Polytechniou 9, Zografou, 157 73 Athens, Greece
akalion@softlab.ntua.gr
[3] Expertnet S.A., Security Research Department, Achilleos 1 Str. & Kifissias 244 Ave.,
Halandri, 152 31 Athens, Greece

Abstract. During the last years, governmental organizations have invested considerable effort and financial resources in the development and adoption of e-government services. In order to sustain the quality of their services, governmental organizations need to solve the problem of efficient and secure electronic exchange and processing of governmental documents and data. A major difficulty in this distributed deployment is the fact that these interconnected systems are heterogeneous and they may operate in multiple organisational domains. This paper demonstrates how the ISO/RM-ODP standard offers a general framework to design and develop an open distributed system attuned to e-government environments. This is subsequently supported by a high level case study of how this standard can be applied in the case of a system designed for small to medium sized European municipalities.

1 Introduction

During the last years, governmental organizations at all levels have invested considerable effort and financial resources in the development and adoption of e-government services. The first era of e-government activities included web site hosting and management. The more sophisticated of them even allowed limited transactions online. The next generation of e-government applications demand is for interactive service delivery, secure transactions, cross-border interactions and a homogenous framework for e-government systems communication. Moreover, the meaningful interaction between governmental organisations in order to provide more sophisticated services to citizens is an ever increasing demand. In order to sustain the quality of their services, governmental organizations need to solve the problem of efficient and secure electronic exchange and processing of governmental documents and data. It is of utmost importance that these services are provided in a way that is easily adoptable and accessible by all citizens, businesses and other public bodies.

Governmental organizations may be distributed within a small area (e.g. a town hall and other municipal offices) or a larger one (across a wider geographical area) and

they have to interact with citizens and other organizations. Therefore, they need distributed system architectures that address all of their specific requirements. The development of a distributed processing system is based on the understanding of the system's functionality and the efficient representation and structure of the system's fundamental information. A major difficulty in this distributed deployment is the fact that interconnected systems are heterogeneous and they may operate in multiple organisational domains.

ISO/RM-ODP (International Standard Organization/Reference Model of Open Distributed Processing) [1], OMG/CORBA (Object Modelling Group/Common Object Request Broker Architecture) [2] and OSF/DCE (Open Software Foundation/Distributed Computing Environment) [3] are examples of standards for open distributed processing that cope with distributed systems heterogeneity and openness. RM-ODP in particular offers a general framework to design and develop an open distributed system and has received an increasing acceptance by the scientific and commercial community over the last years [4], [5], [6], [7]. A recent proposal on an integrated framework for the development of open distributed systems is presented in [8]. This approach is based on UML and the Prototype Verification System (PVS) [9]. It extends the UML notation by assigning formal semantics to the graphical modelling constructs of UML notation. However, it focuses mainly on the system's state transitions, whereas RM-ODP is a more generic framework. Furthermore RM-ODP is closer to human natural language for specifications and thus it is easier to comprehend and use. Therefore, this paper proposes the selection of the RM-ODP standard for the design of an open distributed architecture attuned to e-government. A high level case study of how this standard can be applied in the case of a system for European municipalities is also presented.

The paper is structured as follows: Section 2 gives an overview of the RM-ODP standard and presents its fundamental characteristics; Section 3 presents the specific requirements of governmental organizations that have to be addressed by a system architecture tuned to this environment; Section 4 shows how RM-ODP and its features fulfil these requirements; Section 5 presents a high-level case study of how the RM-ODP standard is being used for the design of a system targeting municipalities and, finally, Section 6 draws conclusions.

2 An Overview of the RM-ODP Standard

ISO and ITU-T, have joined efforts to produce a common framework for developing Open Distributed Processing (ODP) systems that benefit from the distribution of information processing services in environments of heterogeneous technology resources and multiple domains. The ODP Reference Model (RM-ODP) is the result of this effort. RM-ODP creates an architecture that integrates support for distribution, interworking and portability and describes systems that support heterogeneous processing and information exchange between groups within an organization as well as between cooperating organizations [1].

RM-ODP defines the basic concepts of distributed processing, identifies the characteristics that qualify a system as an ODP system and introduces five viewpoints which are used in order to specify an ODP system. A viewpoint on a system is an abstraction of that system (or a part of it) that gives a specification of the whole (or part) of the system related to a particular set of concerns. RM-ODP also defines a viewpoint language that is used to describe each viewpoint. In effect, each viewpoint language provides a set of definitions of concepts and rules enabling the specification of the system from its corresponding viewpoint. Furthermore, RM-ODP provides a framework for checking the system's conformance to the specification and the consistency between the different viewpoints and defines certain functions that are required to support an ODP system. Finally, it presents a system architecture which provides distribution transparencies between system applications. Distribution transparencies enable complexities associated with system distribution to be hidden from applications when they are irrelevant to their purpose [1].

The five viewpoints and their corresponding languages, as defined in RM-ODP, that are used to specify an ODP system are the following:

- Enterprise Viewpoint, a viewpoint of an ODP system and its environment that focuses on the policies which define the behaviour of an object in the system as well as the system's purpose of operation and scope. This viewpoint describes the system from the aspect of what it is required to do. The Enterprise Language is used to describe the Enterprise Viewpoint.
- Information Viewpoint, a viewpoint which specifies and describes the information entities that are communicated, stored and processed in the system. The Information Language is used to describe the Information Viewpoint.
- Computational Viewpoint, a viewpoint which focuses on the way distribution of processing is achieved. The Computational Language is used to describe the Computational Viewpoint.
- Engineering Viewpoint, a viewpoint which focuses on the way different objects of the system use to communicate with each other and the resources that are needed to accomplish this communication. The Engineering Language is used to describe the Engineering Viewpoint.
- Technology Viewpoint, a viewpoint which focuses on the selected technology of a system. The Technology Language is used to describe the Technology Viewpoint.

The system specification of RM-ODP is based on an object modelling approach. This approach provides a formalization of well-established design practices of abstraction and encapsulation.

3 Architectural Requirements of E-government Systems

This section presents the architectural requirements posed by the systems of governmental organizations wishing to engage themselves in e-government.

3.1 Interoperability

It is hard to achieve interoperability in a governmental organization system due to various forms of heterogeneity that exist in governmental environments. The interconnection of governmental organizations that use various platforms and systems is a difficult task requiring easily identifiable and publishable e-services, as well as clear interfaces for the establishment of secure and reliable connection points. Furthermore, even within the boundaries of a single public organization, a system may be spread across a geographical area. For example, in the case of a municipality, the administrative offices may be located further away from the cultural centre or the town hall. The fact that there is a large number of smaller public organizations makes interoperability between their systems even more challenging.

3.2 Scalability

Enhanced scalability is also a requirement that must be met by the infrastructure of governmental organizations due to the large number of citizens that need to be served with acceptable levels of quality of service. Furthermore, public administrations are continuously in the process of deploying electronically traditional paper-based services as well as new services. An e-government system should be able to host an increasing number of e-services.

3.3 Security and Trust

E-government services have to be secure so that government employees and users trust the system and feel confident in using it. Governmental organizations need to design, implement and operate a secure electronic environment for the exchange and processing of governmental e-documents, access to repositories of information for authorized public servants and hosting of shared applications.

3.4 User-Friendliness and Accessibility

A governmental organization environment has to be easily accessible by civil servants and citizens alike, with user-friendly interfaces covering the needs of various types of users. These requirements stem from the fact that governmental organizations have to serve a large number of citizens with diverse information technology training. It is very important that any complex operations be transparent to the end user. A governmental environment should also offer good international support for foreign citizens as well as support for disabled citizens.

3.5 Cost Considerations

Minimization of costs, both of deployment and of operation, is a very important requirement for the successful deployment of e-Government services. This issue is of

even greater importance to smaller public organizations, which frequently do not have the same resources as their larger counterparts, in terms of finances and personnel.

3.6 Transparent Automated Processing

Automated processing is linked to the operational costs of e-Government services for governmental organizations, since they have to satisfy the requests of many people with only a handful of personnel under normal operation.

3.7 Cross-Border Characteristics

An ever increasing number of European citizens change their location to work in other countries. This means that there is a definite demand to support them in administrative procedures that include cross-border communication. In cross-border services there is exchange of information, data or documents between citizens and public administrations (C2G, G2G) in an international context and across administrative boundaries.

3.8 Limited Training

There is always a need for training of the government employees during and after the deployment of e-services. Especially in the case of larger governmental organisations that employ hundreds or thousands of people. Training could prove to be an extremely costly and complicated procedure. The provision of limited required training is of utmost importance for the minimisation of time that has to be invested by the staff before they operate any new system.

3.9 Compatibility with Existing Infrastructures

Citizens and governmental organisations have already established software, hardware and network infrastructures. These infrastructures nevertheless can seldom interoperate within a distributed architecture comprising of several organizations. Therefore, such an architecture must provide a compatibility layer with existing and legacy systems so that their current data can be used, since the complete replacement of legacy systems is usually prohibitively costly.

3.10 Mobility Aspects

Mobile access affords citizens the use of services detached from strict office hours and independent of locations. Mobility and independence add real value to these services. Services with mobile aspects are of interest to facilitate access to the era of mobile Internet services for governmental organizations, which means in particular a device-independent access to services (by cellular phone, PDA, etc.).

4 Suitability of RM-ODP for E-government

The use of RM-ODP as a standard for designing a distributed system enables and supports the development of systems with certain desired characteristics. These characteristics, which are depicted in Table 1 and described below, satisfy all the requirements of a distributed governmental organization system.

Table 1. RM-ODP features against e-Government requirements

	Openness	Integration	Flexibility	Modularity	Federation	Interoperability	QoS	Security	Transparency
Interoperability	√	√			√				
Scalability	√					√	√		
Security/Trust								√	
User Friendliness/Accessibility							√		√
Cost Considerations		√		√					
Transparent Automated Processing			√						√
Cross-border Characteristics	√				√				
Limited Training						√			
Compatibility with Existing Infrastructures		√	√						
Mobility Aspects			√						

4.1 Openness

Openness of a system makes possible the change of the processing node at which a component is situated without the need for modification. This feature supports the scalability needs of governmental organization applications. Such applications must satisfy the demands of a continuously expanding group of citizens and so the machines that are used to host these applications should be upgraded on a regular basis. Openness of the system's design ensures that the migration of the applications to the new hardware and software will be achieved in a smooth fashion.

Openness also ensures the meaningful interaction between components, effectively covering the needs for interoperability and cross-border characteristics. The basis for interoperability is the communication and interaction between the different parts of the system. Furthermore, in order to achieve a satisfying interworking level between the different systems, cross-border characteristics must be integrated into them. The meaningful interaction of all components and the freedom to setup applications at any given node of the system (which are both provided by the system's openness) are essential in satisfying this goal.

4.2 Integration

Integrated systems incorporate various other systems and resources into a whole without costly ad-hoc solutions. Systems with different architectures and different resources are fine-tuned to work together. Integrated systems deal with the problem of heterogeneity and thus satisfy the governmental organization applications' demands for interoperability and compatibility with existing infrastructures. Furthermore, effectively incorporating existing systems and resources into new architectures lowers the cost of the migration procedure to the new solutions. This last feature is of great importance to governmental organizations since their financial resources are very often extremely restricted.

4.3 Flexibility

Flexible systems are capable both of evolving and of accommodating the existence and the continuous operation of existing legacy systems. A major requirement of a governmental organization is the use of the major part of its existing infrastructure when a new IT solution is introduced. Flexibility addresses the requirement of compatibility with existing infrastructure. Furthermore, changes in the system's architecture and topology are easily adopted when a system is flexible. The majority of modern e-Government solutions focus on the interaction with the citizens through more contemporary media (such as palmtop PCs or cellular phones) [7], [10], [11], which is a feature that requires mobile interfaces. Thus, flexibility helps to deal with transparent automated processing and mobility.

4.4 Modularity

The parts of a modular system are autonomous but interrelated. Modularity supports the reuse of previously developed software modules which in turn lowers the cost of a system's development. Governmental organizations can benefit dearly from this option since it comes in terms with their need for low cost solutions, especially when upgrading an existing system or trying to develop a completely new application for newly introduced services.

4.5 Federation

A federated system can be combined with other systems from different administrative or technical domains in order to achieve a single objective. Federated systems are satisfying the needs for interoperability and cross-border characteristics. Oftentimes, a governmental organization's offices and departments are not located in the same building or even the same area however the system of a governmental organization expands to all its premises and its components must interoperate effectively. Federation addresses this need successfully.

4.6 Manageability

ODP based systems are highly manageable. This means that they allow easy monitoring, control and management of resources and processes in order to support configuration, QoS and accounting policies. This directly affects the scalability of the system, since it allows easier management of the increasing number of users and helps with keeping the system operational in spite of the limited personnel training capabilities in governmental organizations.

4.7 Quality of Service (QoS)

Systems designed and implemented based on ODP take into serious account quality of service needs. They intend to cover provision of timeliness, availability and reliability in the context of resources (local and remote) and interactions, together with provision of fault tolerance that allows the remainder of a distributed system to continue to operate in the event of failure of some part. This means that ODP systems offer high availability, dependability and accessibility, which is very important when the system has to be continuously up and running to cover the diverse needs of a large number of citizens throughout the day. The QoS directly affects how users perceive the interaction with the services.

4.8 Security

ODP proposes a strong security framework for architectures, ensuring that system facilities and data are protected against unauthorized access and respect privacy. Certain functions are defined that are fundamental and widely applicable to the development of ODP systems. More specifically, ODP defines security functions (such as access control function, authentication function, integrity function and key-management function) which help organize and orchestrate the development and application of security policies within an ODP system. Security requirements posed by governmental organizations are generally strict and they are made often more difficult to meet in modern governmental infrastructures by the increasing needs for remoteness of interactions and mobility of parts of the system and of the system users.

4.9 Transparency

ODP has as primary goal the integration of transparency in systems. Transparency hides the details and differences in mechanisms used to overcome problems caused by distribution of applications. Aspects of distribution which should be masked (totally or partially) include: heterogeneity of supporting software and hardware, location and mobility of components and mechanisms to achieve the required level of QoS in the face of failures (e.g. replication, migration, checkpointing, etc.). RM-ODP introduces a set of distribution transparencies in order to support this need. Transparency is critical for a number of governmental organization requirements: user friendliness in order to hide application and processing details from users and automated processing so

that a small number of civil servants and public workers can manage systems successfully.

5 The eMayor Case Study

The practical suitability of RM-ODP in e-government is demonstrated in this paper through an actual implementation. In the IST eMayor project [12], we have opted to use RM-ODP in the design and implementation of an e-government platform attuned to the environment of European Small to Medium Sized Governmental Organizations (SMGOs) [13], which in the eMayor case include several European municipalities. During the design phase we have applied RM-ODP concepts and principles in combination with UML to define a system that addresses the various needs of the municipalities, as they have been identified during the analysis and user requirements collection phase of the project [14]. This section presents the initial design concepts we have used in eMayor with examples from the first three RM-ODP viewpoints (Enterprise, Information and Computational).

As part of the Enterprise Viewpoint, eMayor has identified the main stakeholders in the municipality community domain and the roles they assume in the environment, such as the citizens and their delegates and the civil servants working in the municipalities or other organizations. Furthermore, we have identified the business processes that take place in the communities along with the activities of the various roles in them and we have described the policies and constraints related to the processes. This has led to the definition of Enterprise objects. One example is the PKI sub-community within the wider eMayor community that comprises the Regional CAs, a Pan-European CA, a Pan-European Validation Authority and the various registered Users (registered Civil Servant etc.) as shown in figure 1:

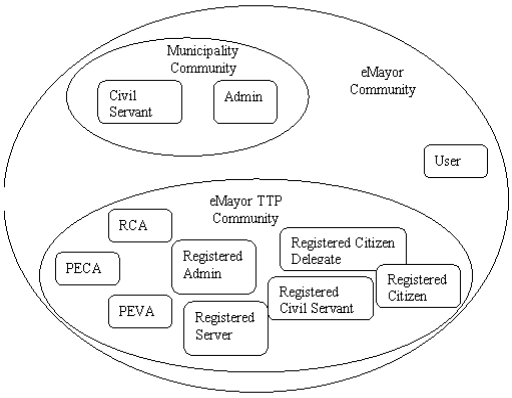

Fig. 1. The eMayor TTP community

The Enterprise Viewpoint defines all relationships among the entities and the processes they follow, e.g. for the PKI framework setup of an end-entity or the Certificate Validation. The specifications utilize free-form diagrams as well as UML diagrams (Use Case, Class, Collaboration and Activity diagrams) for the representation of the above concepts.

The Information Viewpoint has presented and analyzed various Information Objects that will be used by the eMayor platforms and communicated between them. Examples of such information objects are the User Credentials, User Profiles, Service Requests, a Registry etc. Class diagrams are used in order to represent the Information Objects, such as the one in figure 2 for the Service Request Information Object:

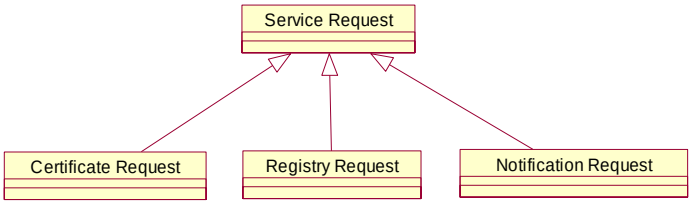

Fig. 2. Service Request Information Object

The Service Request object represents requests made for a specific service offered by the eMayor system such as certificate issuance, registration to the taxes services or a notification about the state of processing of another request. As can be seen in Figure 2, the subclasses of the Service Request Information Object are: Certificate Request, Registry Request and Notification Request. Information Objects pass from various states as part of their life-cycle and the state transitions of the Information Objects are depicted in the Information Viewpoint using UML StateChart diagrams.

The Computational Viewpoint demonstrates the overall architecture of the eMayor system, divides the computational functionalities in distinct packages and depicts their interconnection and collaboration. The packages are elaborated with further details showing the Interfaces exposed by each package, as well as the internal mechanisms and modules. The computational packages identified so far are User Interface, Output Processing, Policy Enforcement, Format Transformation, Content Routing, Service Handling, Legacy Systems Adaptation and Persistent Storage as shown in figure 3:

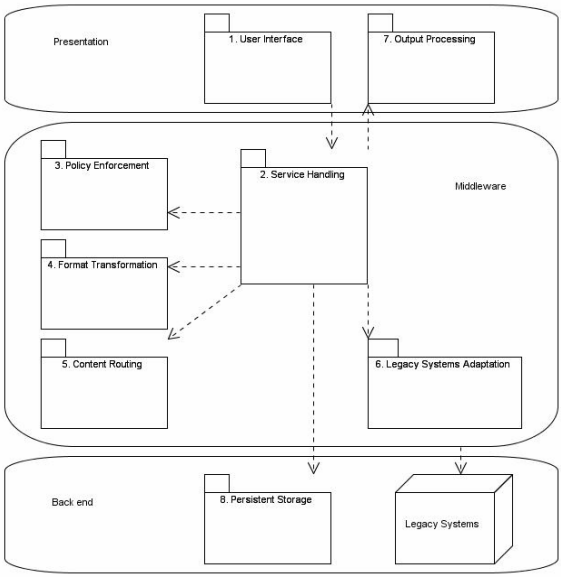

Fig. 3. The eMayor system architecture

The User Interface communicates with the Service Handling for the actual processing of the service and with the Policy Enforcement for access control, the security mechanisms and the policies of the municipalities' legal frameworks. Service Handling represents the core of the system and has dependencies to all other packages. It communicates with the Policy Enforcement (e.g. for access control, encryption and digital signature of documents and messages, etc.). The Format Transformation is responsible for transforming legal documents from a country-bound local format to a universal format for transport within the eMayor environment and vice versa. Content Routing provides the routing functionality for forwarding requests and legal documents to another municipality. The Legacy Systems Adaptation is the linking point with the legacy systems of the municipalities. The Persistent Storage modules handle storage to the file system or databases. Finally, Output Processing provides support for outputting information to devices, printing and notification services.

The specifications are supported by UML Component and Class diagrams for the presentation of Interfaces and Classes and Sequence diagrams for the presentation of sequences of actions involving multiple computational components.

6 Conclusions

In this paper we have demonstrated the suitability of the RM-ODP standard in the design of open distributed architectures attuned to e-government environments. The use of RM-ODP as the preferred framework encourages a clear separation of concerns

(through the different viewpoint specifications), which in turn leads to a better understanding of the problems being addressed during the design of such a system. In the context of the eMayor project, we have applied RM-ODP concepts in order to design an e-government platform covering the needs of small to medium sized European municipalities. This paper has presented the initial status of the work towards this direction by giving a short overview of the first three viewpoints of the specification.

7 Acknowledgements

The authors would like to thank the E.C. for its support in funding the eMayor project (IST-2004-507217) and all the members of the project consortium for valuable discussions.

References

1. ITU-T Rec. X.901 | ISO/IEC 10746-1,2,3, “Reference Model for Open Distributed Processing – Part 1: Overview, Part 2 Foundations, Part 3 Architecture”, 1996-98
2. Object Modelling Group/Common Object Request Broker Architecture, http://www.omg.org
3. Open Software Foundations / Distributed Computing Environment, http://www.osf.org
4. C.A. Costa, J.A. Harding, R.I.M Young, “The application of UML and an open distributed process framework to information system design”, Computers in Industry, Elsevier Science, 2001, pp. 33-48
5. M. A. Nankman, L. J.M. Nieuwenhuis, “Specification of a distributed storage system”, Computer Communications, Elsevier Science, 1996, pp. 30-38
6. M. Blinov, A. Patel, “An application of the reference model for open distributed processing to electronic brokerage”, Computer Standards and Interfaces, Elsevier Science, 2003, pp. 411-425
7. German Federal Ministry of Interior, “SAGA - Standards and Architectures for e-government Applications, version 2.0”, December 2003
8. I. Traore, D. Aredo, H. Ye, “An integrated framework for formal development of open distributed systems”, Information and Software Technology, Elsevier Science, 2004, pp. 281-286
9. S. Owre, N. Sankar, J. Rushby, D. W. Stringer-Calvert, “PVS Language Reference v. 2.3”, September 1999
10. Project “eGovernment for Development”, 2004, http://www.egov4dev.org/topic4.htm
11. National Electronic Commerce Coordinating Council (NECCC), “M-Government: The Convergence of Wireless Technologies and e-Government”, 2004, http://www.ec3.org/Downloads/2001/m-Government_ED.pdf
12. E.C. 6th Framework Programme, Electronic and Secure Municipal Administration for European Citizens – eMayor, IST-2004-507217, 2004. Available form: www.emayor.org
13. A.Kaliontzoglou, P. Sklavos, T. Karantjias, D. Polemi, “A secure e-Gorvenment platform architecture for small to medium sized public organizations”, Electronic Commerce Research & Applications, Elsevier Science, 2004 (Article in Press)
14. eMayor consortium, “Deliverable D2.1: Municipal Services – Analysis, Requirements and Usage scenarios”, eMayor project, IST-2004-507217, 2004, available from www.emayor.org