E-Government: on the Way Towards Frameworks for Application Engineering –draft–

Marie-Noëlle TERRASSE[1,4,6], Marinette SAVONNET[1,6], Eric LECLERCQ[1,6],
George BECKER[2,6], Thierry GRISON[1],
Laurence FAVIER[4,5], and Carlo DAFFARA[3,6]
(1) LE2I (UMR CNRS 5158), University of Burgundy,
E-mail: firstname.lastname@u-bourgogne.fr -
(2) E-mail: gbecker@nerim.net -
(3) Conecta Telematica, Italy, E-mail: cdaffara@conecta.it -
(4) Pr@tsic, Maison des Sciences de l'Homme, University of Burgundy -
(5) Centre Georges Chevrier (UMR CNRS 5605), University of Burgundy,
E-mail: laurence.favier@u-bourgogne.fr - (6) OpenTTT European project

Grants to the OpenTTT European project (INN7-030595): "**O**pen source **P**artnership for **E**nterprises and **N**etwork of developers aiming at **T**ransnational **T**echnology **T**ransfer".

Abstract. In this article we present high-level architectures for e-Government applications. These architectures depend on a country's strategy for e-Government integration and they give rise to two major issues. The first issue is how to guarantee semantical quality of information regardless of the chosen architecture. The second issue is how to facilitate sound transition of e-Government applications from one architecture to another under evolutionary pressures of a country's political strategy. In order to address these two issues we use Model-Driven Engineering which places metamodels, models and their transformations at the core of the engineering process. Overall semantical quality is thus guaranteed by metamodels while model transformations guarantee soundness under evolution.

We propose two adjustments to OMG's architectures for Model-Driven Engineering of highly-complex application domains. In OMG's architectures, a metamodel describes an application domain (reusable information) while a model describes an application (contextual information). By introducing a reusable model for a family of applications, we can share pieces of model-level information.

1 Introduction

E-Government applications should be able to evolve incrementally since they belong to a relatively stable domain. Legacy information systems of public administrations operate in well-known domains. They generally rely on stable and recognized vocabularies and they are used in the context of unchanging business processes. Yet, the spreading of new technologies and the expectations of various actors (citizens, administrative project leaders, politicians) push towards development of innovative information systems. In fact, E-Governement implies several major changes in administration business processes:

- a citizen-centered approach to e-Government which is based on availability of services dedicated to life and business events (e.g., birth, marriage, as well as setting up a company, paying taxes, participating in procurement activities) and delivered through various channels [7, 1];

- a separate management of services and their delivery through multi-channel portals;
- an integration of administration services with respect to national strategies and citizens' expectations, administrative staffs' working habits, and international strategies.

Even though administrative portals are the most visible part of current developments, E-Government's integrated services are not restricted to front-office evolution. Back-office reorganization [5, 7] in turn makes it necessary to harmonize and to make consistent all levels of administration: local, national/federal, international (e.g., pan-european services) in order to enable interoperability of e-Government information systems. Such interoperability is rather difficult to set up, since e-Government applications generally exhibit strong heterogeneities, such as data heterogeneity (formats ranging from alpha-numeric data to cadastral map images, quality, semantics), actor heterogeneity (members of various administrations, end-users, or politicians which are given authorizations to access data and to use services), and heterogeneity of applications' objectives.

Furthermore, e-Government applications generally do not have precise non-functional specifications (such as those regarding security, confidentiality, and performance) even though many interoperability domain-dedicated frameworks that have been built recently enable e-signature, personal identification and exchange of data between administrations, (e.g., IETF, OASIS, WS-I, UNCEFACT, e-GIF, OOI, RGI [10, 11, 12, 13, 14, 15]). Such domain-dedicated frameworks can be used together with technical specifications and architecture components [6, 9] that were offered to web-enabled application designers either by an international consortium, or by national structures (e.g., the Security Assertion Markup Language, the Identity Federation Framework that provides *Single Sign On* facilities, the UN/CEFACT Modeling Methodology, and the COSPA project [16, 17]).

Most E-government applications can be described in terms of a loosely coupled integration of administrative information systems (from various administrations) to which up to three extra components can be added. The first component provides *core business integration*, i.e., it enables data and process consolidation. The second component is a *portal for administrative staff members* providing a unified access to information and services of each administration. The third component is a *portal for end-users* that offers an integrated view of all administrations regardless of their actual organization. Depending on the chosen components we define an architecture schema which we call an *application profile*. We propose eight different application profiles, and present them in Figure 1.

The technical aspects of e-Government applications show that various basic components are necessary. For example, end-user portals should rely on an identity federation framework while administrative portals should encompass a language for expressing security and authorization rules. Similarly, the integrated core business should rely on knowledge and business process descriptions (e.g., ontologies, metamodels, models [2, 4]).

We define four basic rules for selection of framework components. First, end-user portals are supposed to federate identities from various legacy e-Government applications. Second, administrative portals must encompass authorization descriptions and enforcement, as well as common vocabularies (formulated in terms of shared ontologies). Third, core business integration cannot be carried out without at least a common vocabulary (formulated in terms of shared ontologies). Fourth, each architecture must include security components.

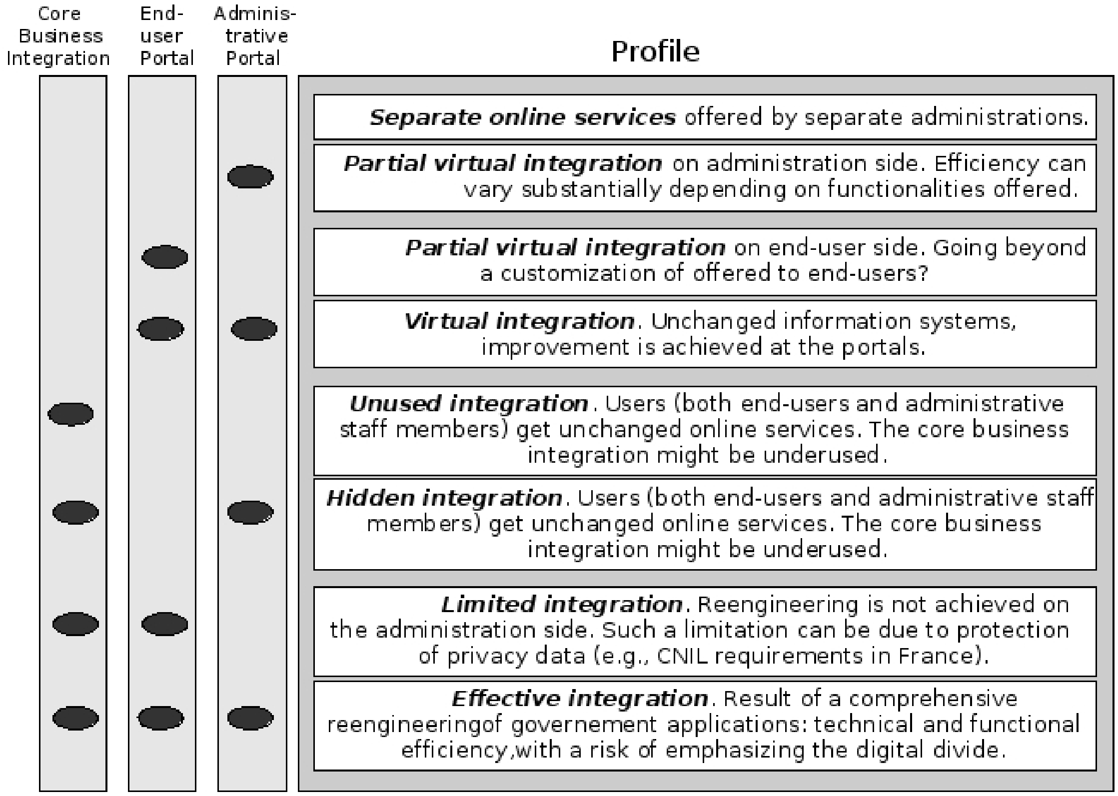


Figure 1: Different profiles of e-Government applications.

2 An MDE perspective on E-Government applications

OMG's metamodeling architectures strive to structure an application description into four levels: instance, model, metamodel, and meta-metamodel. The meta-metamodel level describes how the real world is seen, which high-level languages are used to describe the real world (e.g., description of a semantics of space and time). The metamodel level defines which language will be used for modeling of a specific application domain (e.g., a metamodel extended with constructs for spatio-temporal descriptions). The model level describes a given application (e.g., a model of a GIS for state and territorial border management). The instance level contains objects which belong to such a GIS (e.g., the French-German border after World War I, the border between the Brooklyn and Staten Island boroughs in New-York in 1964).

Metamodels that were originally introduced as languages for model description [3] turned into languages for application domain descriptions (Domain Specific Languages [8]). Reuse is the key concept for application domain descriptions. MDE expresses such reuse at the metamodel level. Yet, building a model from a metamodel in case of a complex application requires a huge amount of work. We desire to reuse part of the modeling work: we thus propose to describe a family of applications in terms of a reusable model. A definition of such a reusable model distinguishes *abstraction separation* between metamodels and models (Figure 2.a) from *methodological separation* between reuse and contextualization (Figure 2.b). Each specific application is then built as a specific instance of the reusable model. Figure 2.c presents an example metamodel and two reusable models for e-Government applications with two different types of confidentiality requirements.

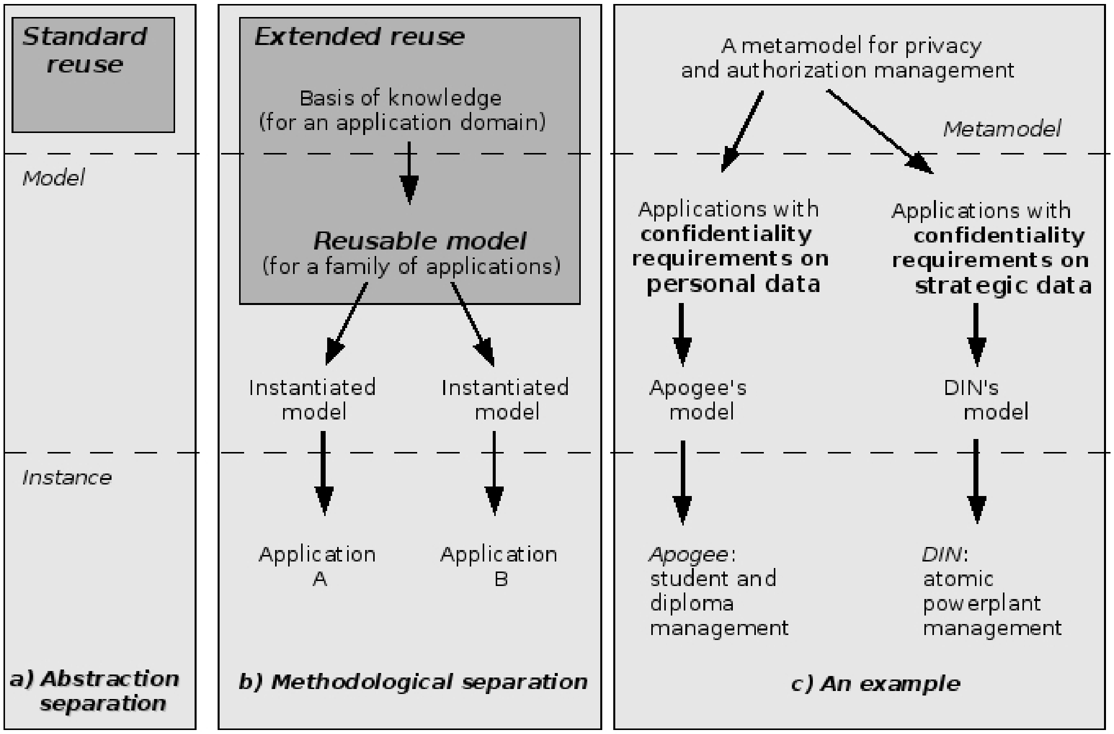


Figure 2: Metamodeling levels and reuse boundary

3 Illustrative Example: Metamodeling a Data Protection Strategy

Enforcing data protection policies in order to satisfy legal and security requirements is a major issue for e-Government applications. In order to keep our example reasonably small, we limit ourselves to a simplified context. We use the following vocabulary. E-Government applications use *resources* which are mainly documents containing *data*. Data elaboration is limited to two categories: *raw* data (e.g., the grades obtained by a student) and *aggregate* data (e.g., yearly averages of student grades). Depending on the data they contain, resources are classified according to the level of data *protection* they require. Data protection falls under three categories: *public*, *confidential*, and *private* data. Public data can be read by everybody (e.g., a list of the diploma delivered by a university), access to confidential data is restricted to administrative staff members (e.g., the grades obtained by students), private data can only be read by specially authorized administrative staff members (e.g., medical record for a disabled student). In order to manage access to private data, *authorizations* are delivered either on an *individual* basis or *statutorily*. Statutory authorizations are delivered by an administration to its staff members. Individual authorizations are delivered under responsability of *authorization granting authorities*.

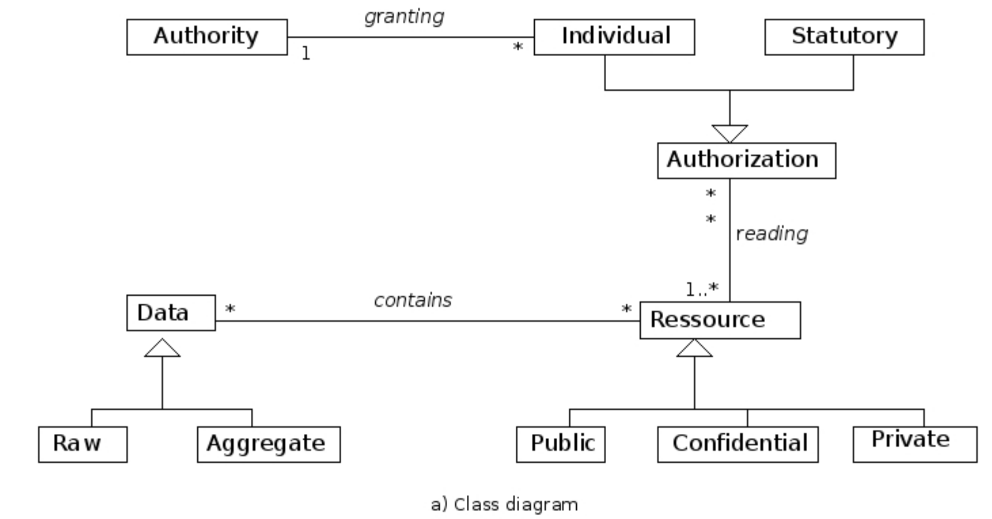
Introductory model Let us consider the general case where the following rules apply:

R1- Public resources cannot be associated with authorizations.

R2- Resources containing only aggregate data cannot be private.

R3- Confidential resources must be associated with authorizations.

Figure 3 presents an introductory model of e-Government applications in terms of a UML class diagram. Resources and data are represented by classes. The class *Resource* is specialized into classes *Private*, *Confidential*, and *Public*. The class *Data* is specialized into classes *Raw* and *Aggregate*. Authorizations are represented by a




```
Context Public
inv r2: self.reading -> isEmpty()

Context Ressource
inv r3:  ( self.contains -> forAll (d : data | d.oclIsTypeOf(Aggregate))
            implies not self.oclIsTypeOf(Private)
```

b) OCL constraints (class invariants)

Figure 3: Data protection policy: introductory model (class diagram)

class *Authorization* together with two specialized classes *Individual* and *Statutory*. Authorization granting authorities are represented by a class *Authority*. An association, called *reading*, links resources with authorizations. An association, called *granting*, links individual authorizations with granting authorities. In order to guarantee modeling accuracy, it is necessary to make sure that rules R1 to R3 are expressed in the model. Rule R1 can be expressed as a specialization of the *reading* association. This specialization links *Confidential* with *Authorization* and has multiplicity set to 1..* at the *Authorization* end. Rules R2 and R3 must be expressed in the form of OCL constraints (e.g, as invariants of the class *Public* and *Resource*, respectively). These two rules are given in Figure 3.b.

Metamodel In order to express domain-related knowledge at the metamodel level, we define three major concepts within the application domain, namely data, resources, and authorizations together with their relations. We then define five stereotypes: a stereotype ≪*D*≫ for modeling data, a stereotype ≪*R*≫ for resource modeling and expressing rules R1 and R2, a stereotype ≪*A*≫ for modeling authorizations, a stereotype ≪*RD*≫ for modeling reading, a stereotype ≪*RA*≫ for grant modeling.

We choose to express rule R3 within reusable models since it pertains to the set of data elements associated with a resource. Figure 4 depicts the proposed metamodel: in part a) the proposed stereotypes are depicted in light gray, part b) presents the OCL expression of constraints c1 and c2 (which express rules R1 and R2, respectively).

Reusable models By using the above metamodel, we define two example reusable models corresponding to two families of applications that share the same data protection policy. The corresponding reusable models are given in Figure 5.

Our first example is a family of applications centered on protection of personal data. In such applications authorizations are statutory (invariant *r4* of the class *Authorization*), and confidential resources must be associated with authorizations (invariant *r5* of the

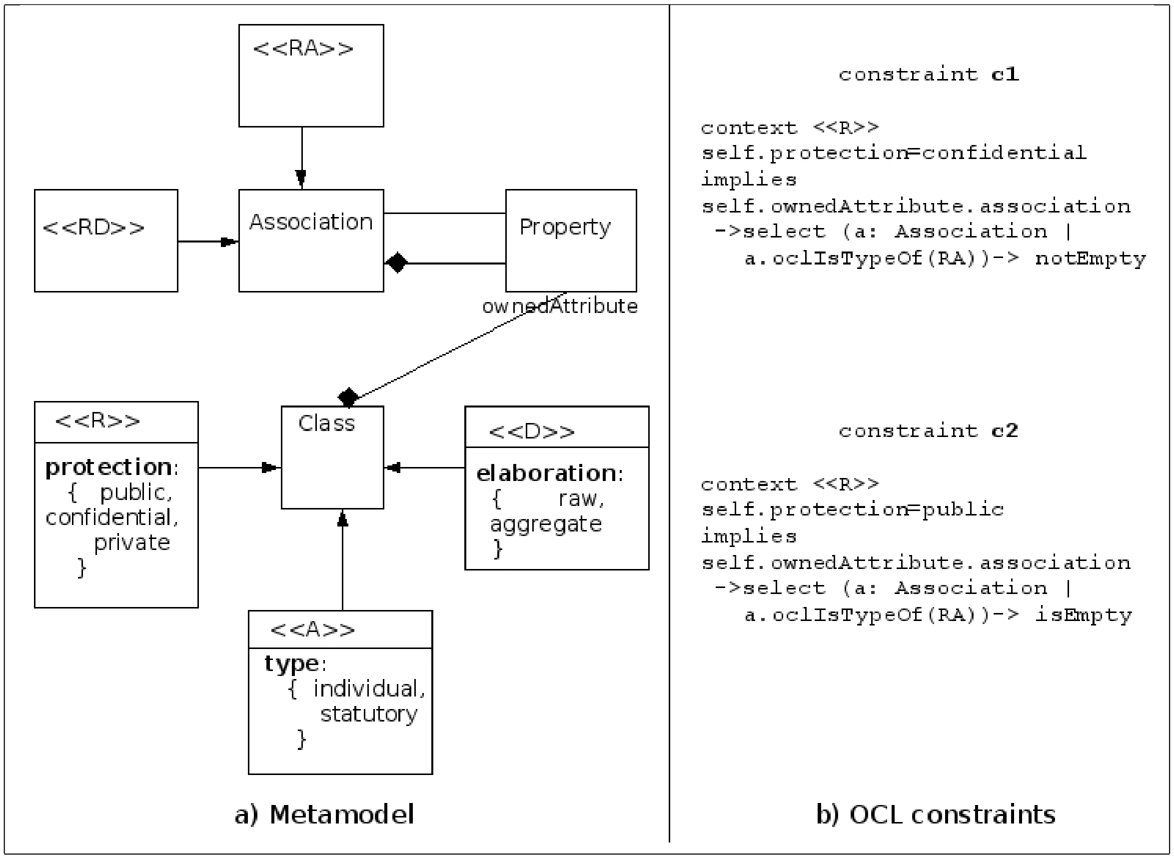


Figure 4: Data protection policy: domain-metamodel for e-governement applications with confidentiality requirements

class *Resource*). Rule R3[1] must be enforced (invariant *r6* of the class *Resource*).

Our second example is a family of applications centered on protection of strategic data. In such applications, confidential resources must be associated with individual authorizations (invariant *r8* of the class *Resource*) and aggregate data are not necessarily public though they may not be private (invariant *r7* of the class *Resource*). Rule R3 is subsumed by invariant *r7*.

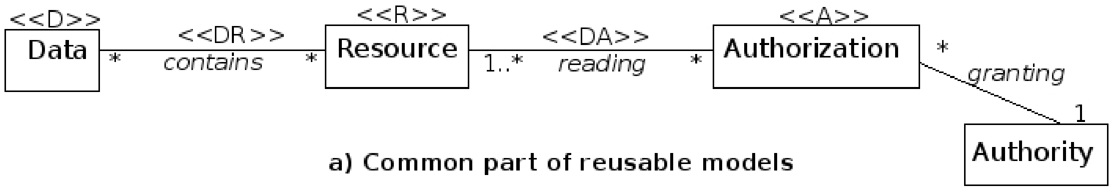
As stated in the above sections, reusable models allow reuse within families of e-Government applications. One of the major challenges is to appropriately define such families, which is particularly difficult for highly complex application domains. We propose to use each of the profiles of e-Government applications from Figure 1 as a family. A reusable model thus describes bases on which the integrated core business and portals of an application profile can be built.

The benefit that we obtain is sound evolution of e-Government applications from one profile to another since models can be transformed under the control of the shared metamodel and of their source and target reusable models. Furthermore, satisfactory semantical quality of each model can be guaranteed (by means of a reference metamodel and reusable model).

4 Conclusion

In this paper, we have discussed the possible architectures of e-government applications. Two majors requirements apply to such architectures. First, these architectures have to enable sophisticated interoperability of legacy applications. Beyond integration of business processes

[1]Rule R3: Resources containing only aggregate data cannot be private.



a) Common part of reusable models

```
Context Authorization
inv r4: self.oclIsTypeOf(statutory)

Context Ressource
inv r5: self.oclIsTypeOf(confidential) implies self.reading-> notEmpty()

inv r6: ( self.contains->notEmpty() and
          self.contains->select(d: data | d.oclIsTypeOf(raw))-> isEmpty()
        ) implies self.oclTypeOf(public)
```

b) Additional constraints for reusable model of personal data

```
Context Ressource
inv r7: ( self.contains->select(d: data | d.oclIsTypeOf(raw))-> isEmpty()
        ) implies not self.oclTypeOf(private)

inv r8 : self.oclIsTypeOf(confidential) implies
         ( self.reading->notEmpty() and
           self.reading->select (a: authorization | a.oclIsTypeOf(statutory))
          ->isEmpty()
```

c) Additional constraints for reusable model of strategic data

Figure 5: Data protection policy: reusable models for e-Governement applications with confidentiality requirements

and information, interoperation of e-government applications must enable: 1) end-users to obtain services from the integrated system for their life events. 2) administration staff members to access and control information and services.

Second, sound evolution of an e-government application architecture must be guaranteed so that the architecture conforms to the country/region/administration strategy.

In order to satisfy the two above requirements, an MDE perspective on e-government has been introduced. Architectures of e-Government applications are thus described in terms of metamodels and models. In order to emphasize model-level reuse and semantical quality of the integrated information, we have described families of applications in terms of reusable models.

Choosing characteristics of families of applications that are described by reusable models is a major issue for improvement of model-level reuse. Our on-going work is to validate the criteria defined in this paper, namely data protection strategies (a family of applications is defined by a given data protection strategy). In order to perform such a validation, various experiments will be carried out (including a one-year experiment with the French health care system.

References

[1] E-government strategy 2002. Technical report, Executive Office of the President Office of Management and Budget Washington D.C. 20503, 2002.

[2] C. Atkinson and T. Khne. Model-Driven Development: A Metamodeling Foundation. *IEEE Software*, 20(5), 2003.

[3] Colin Atkinson. Meta-Modeling for Distributed Object Environments. In *First International Workshop on Enterprise Distributed Object Computing, EDOC'97*, pages 90–101. IEEE, October 1997.

[4] G. Brunet, M. Chechik, S. Easterbrook, S. Nejati, N. Niu, and M. Sabetzadeh. A Manifesto for Model Merging. In *Proceedings of the 1st ICSE Int. Workshop on Global Integrated Model Management*, 2006. China.

[5] Reorganization of Back-offices for Better Electronic Public Services – European Good Practices. Technical report, Danish Technological Institute & Institut für Informationsmanagement, Bremen, 2004. Volume 1: main 15.

[6] B. Elvesæter, A. Hahn, A.J. Berre, and T. Neple. Towards an Interoperability Framework for Model-Driven Development of Software Systems. In *Proceedings of the 1st Int. Conf. on Interoperability of Enterprise Software and Applications, Switzerland*, 2005.

[7] Interoperability for Pan-European e-Government Services. Technical report, European Union. COM(2006) 45 final, February 13, 2006.

[8] M. Mernik, J. Heering, and A.M. Sloane. When and How to Develop Domain-Specific Languages. *ACM Computing Surveys*, 37(4), 2005.

[9] M. Soden, H. Eichler, and J. Hoessler. Inside MDA: Mapping MOF 2.0 Models to Components. In *Proceedings of the First European Workshop on Model Driven Architecture with Emphasis on Industrial Application, University of Twente, The Netherlands*, 2004. Available at URL `http://modeldrivenarchitecture.esi.es/mda\_workshop.html`.

[10] The Internet Engineering Task Force (IETF). `www.ietf.org`.

[11] OASIS. `www.oasis-open.org`.

[12] Web Services-Interoperability Organisation (WS-I). `www.ws-i.org`.

[13] e-GIF. `www.govtalk.gov.uk`.

[14] OOI. `http://standarder.oio.dk/English/`.

[15] General Interoperability Reference (RGI). `ww.adele.gouv.fr/article.php3?id\_article=1064`.

[16] United Nations Centre for Trade Facilitation and Electronic Business (UNCEFACT). `www.ebxml.eu.org/default.htm`.

[17] COSPA Project. `http://www.cospa-project.org`.