E-Government: on the way towards frameworks for application engineering


- **OpenTTT**: a European project **INN7-030595**
  “Open source Partnership for Enterprises and Network of developers aiming at Transnational Technology Transfer” (in four business areas including E-government).

- **Pratsic**: a cross-disciplinary research group on E-government at the University of Burgundy
Introduction on E-government applications

- **Stability**
  - Legacy information systems & well-known domains
  - Recognized vocabularies
  - Stable business processes

- **versus changes**
  - A citizen-centered approach with services:
    - dedicated to life-events, business-events
    - delivered through various channels
  - An integration of administration services:
    local, national/federal, pan-european, international
Security, confidentiality, performance:

- Precise **non-functional specifications** exist (few)
- **Domain-dedicated frameworks** exist (many)
  - e-signature, personal identification
  - exchange of data between administrations

E.g., IETF, OASIS, WS-I, UNCEFACT, e-GIF, OOI, RGI

Security Assertion Markup Language, the Identity Federation Framework (Single Sign On)
Proposed methodology

The methodology we propose for engineering of e-governement information systems consists of four steps:

- Defining possible architectures of such an IS
- Make IS evolve by moving from one architecture to another (component approach)

- Identifying domain-related non-functional properties
- Expressing such properties in a metamodeling architecture: in terms of metamodels and reusable models
A generic structure of E-government applications

- loosely coupled cooperation
- three optional components
  portals
  core business integration

Optional core business integration

Optional end-user portal

Loosely coupled cooperation

Optional administrative portal
## Profiles of E-government applications

<table>
<thead>
<tr>
<th>Core business integration</th>
<th>Optional end-user portal</th>
<th>Adm. portal</th>
<th>Integration type</th>
<th>Components</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>No integration</td>
<td>Security</td>
</tr>
<tr>
<td></td>
<td></td>
<td>●</td>
<td>Administrative virtual integration</td>
<td>Authorization and security, Ontologies</td>
</tr>
<tr>
<td></td>
<td>●</td>
<td>●</td>
<td>End-user virtual integration</td>
<td>Security and identification</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Virtual integration</td>
<td>Authorization, security, Ontologies, identification</td>
</tr>
<tr>
<td>●</td>
<td></td>
<td></td>
<td>Unused integration</td>
<td>Ontologies, security</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Hidden integration</td>
<td>Authorization, identification and security, Ontologies and processes</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Limited integration</td>
<td>Identification and security, Ontologies</td>
</tr>
<tr>
<td>●</td>
<td>●</td>
<td>●</td>
<td>Effective integration</td>
<td>Authorization, identification and security, Ontologies and processes</td>
</tr>
</tbody>
</table>
Encountered problems

- How to move from one profile to another:
  - by the component paradigm
  - by adding and removing components

- How to guarantee global properties:
  - by combining components
  - throughout component changes

- How to emphasize reuse:
  - at the business component level
  - at the non-functional aspects level

We propose **an MDE approach**

based on the OMG’s metamodeling architecture
The metamodeling architecture (OMG)

Four abstraction levels:

- **meta-metamodel**: how the real world is seen
  *a semantics of space and time*

- **metamodel** level:
  a language for modeling of an application domain
  *constructs for spatio-temporal descriptions*

- **model** level: a given application
  *a model of a GIS for state and territorial border management*

- **instance level**:
  *the border between the Brooklyn and Staten Island boroughs, NY, 1964*
An **additional level for strengthening reuse:**

- meta-metamodel
- **metamodel** level:
  a language for modeling of an application domain
- **reusable model** level:
  a partial description (for a family of applications)
- **model** level: a given application
- instance level
Extended reuse

**Standard reuse**

**Extended reuse**

Basis of knowledge (for an application domain)

**Reusable model**
(for a family of applications)

**Metamodel**

A metamodel for privacy and authorization management

Applications with confidentiality requirements on personal data

Applications with confidentiality requirements on strategic data

**Model**

*Instantiated model*

Application A

Application B

**Instance**

*a) Abstraction separation*

*b) Methodological separation*

**c) An example**

Apogee's model

DIN's model

Apogee: student and diploma management

DIN: atomic powerplant management
Introductory model for our example

- **R1**: Public resources cannot be associated with authorizations (*OCL constraint*).
- **R2**: Resources containing only aggregate data cannot be private (*OCL constraint*).
- **R3**: Confidential resources must be associated with authorizations (*Specialization of the reading association*).
Metamodel for our example

- association stereotypes
- class stereotypes
  - ressources
  - data
  - authorizations

Univ. Burgundy – LE2I UMR CNRS 5158 – France – 12
Reusable models for our example

- The common part of reusable models

  ![Diagram of data, resource, and authorization relationships]

- Additional constraints for **personal data**
  - authorizations are statutory
  - confidential resources must have **authorizations**
  - resources not containing raw data must be public

- Additional constraints for **strategic data**
  - confidential resources must have **individual authorizations**
  - resources with aggregate data may be public or confidential
On-going work

- **Testing of our methodology**
  One-year experiment
  Eric Lamy, CNAM (Caisse Nationale Assurance Maladie)
  2007-2011 development plan of CNAM:
  citizen-oriented web services

- To validate **data protection strategies**
  as high-level criteria
  for e-administration systems

- To elaborate a methodology
  for **choosing model-level reusable elements**
  in a given application domain