E-forms services for the Public Sector: Shifting Development Effort to Domain Experts

Stelios GORILAS[1], Kostas VASSILAKIS[2], Tomás Pariente LOBO[3], Efthimios TAMBOURIS[1]
[1]*Archetypon S.A, 236 Sygrou Avenue, 176 72 Athens, Greece*
Tel: +30 2109536410, Fax: +30 2109536411, Email: {sgorilas,tambouris@archetypon.gr}
[2]*e-Gov Lab, University of Athens, 157 84 Athens, Greece*
Tel: +30 2107275220 , Fax: +30 2107275214 , Email: costas@di.uoa.gr
[3]*INDRA, Avda de Bruselas 35, 28108 Arroyo de la Vega, Alcobendas, Madrid, Espana*
Tel: +34 914809140 , Fax: +34 914809161 , Email: tpariente@indra.es

Abstract: E-forms have a central role in a significant number of e-government services. This paper presents a knowledge-based technical platform aiming to assist public sector employees to generate online transaction services by simplifying their development, maintenance and integration with installed IT systems. At the heart of this platform lies the knowledge and transaction services repository. This repository consists of a number of XML document types that incorporate all necessary details for creating and managing online transaction services. The main underlying idea is to provide a platform with intuitive interfaces that can be used directly by domain experts thus minimising the need for personnel with IT skills. This platform is currently under development within the IST SmartGov project.

1. Introduction

According to the European Commission “transaction services, such as electronic forms, are perceived as the future of electronic government” [1]. However, in Europe the potential of these services has not been yet realised by the public sector. Public administrations have so far failed to exploit the benefits of using online transaction services, such as e-forms, in their processes. Although a large number of initiatives have been undertaken at a local, regional or even national level, they have not yet provided the expected results.

SmartGov is a two-year research and development project aiming to specify, develop, deploy and evaluate a holistic approach for online transaction services specific to the public sector [2] [3]. To achieve this objective, SmartGov develops a knowledge-based technical platform to assist public sector employees to generate online transaction services by simplifying their development, maintenance and integration with installed IT systems.

The aim of this paper is to present an overview of the SmartGov technical platform. Therefore, the related re-engineering models at the public authorities as well as the social aspects associated with the SmartGov approach, although part of the SmartGov project, are outside the scope of this paper.

This paper is organised as follows. In section 2, the main objectives of this study are outlined and the methodology used is presented. In section 3 the technological architecture of the SmartGov platform is outlined. In section 4 one of the most important components of the platform, namely the knowledge and transaction services repositories, in more detail discussed. In section 5 the main results achieved so far, the next steps and the business benefits of the SmartGov platform are briefly presented. Finally, in section 6 the main conclusions are given.

2. Objectives and Methodology

The main objective of the SmartGov technical platform is to enable public authorities to create e-forms services on their own, by exploiting the knowledge that lies within the organisation in tacit or explicit form. The platform capitalises on knowledge management principles in order to encompass the knowledge and insight that exists in a public domain to support the development of transaction services. This platform is expected to shift in a high degree the burden of developing e-forms from software engineers to domain experts.

Technically, the development of the technical platform follows the Rational Unified Process [4][5]. This approach states the software engineering process as an incremental iterative process and uses the new standard: Unified Modelling Language. The incremental iterative process means that phases of development are performed not strictly sequentially, but rather they are partially overlapping. The project life is divided into small phases, which have to be refined with the results of previous phases.

3. SmartGov Technical Architecture

In the following figure the high-level architecture of the SmartGov platform is presented.

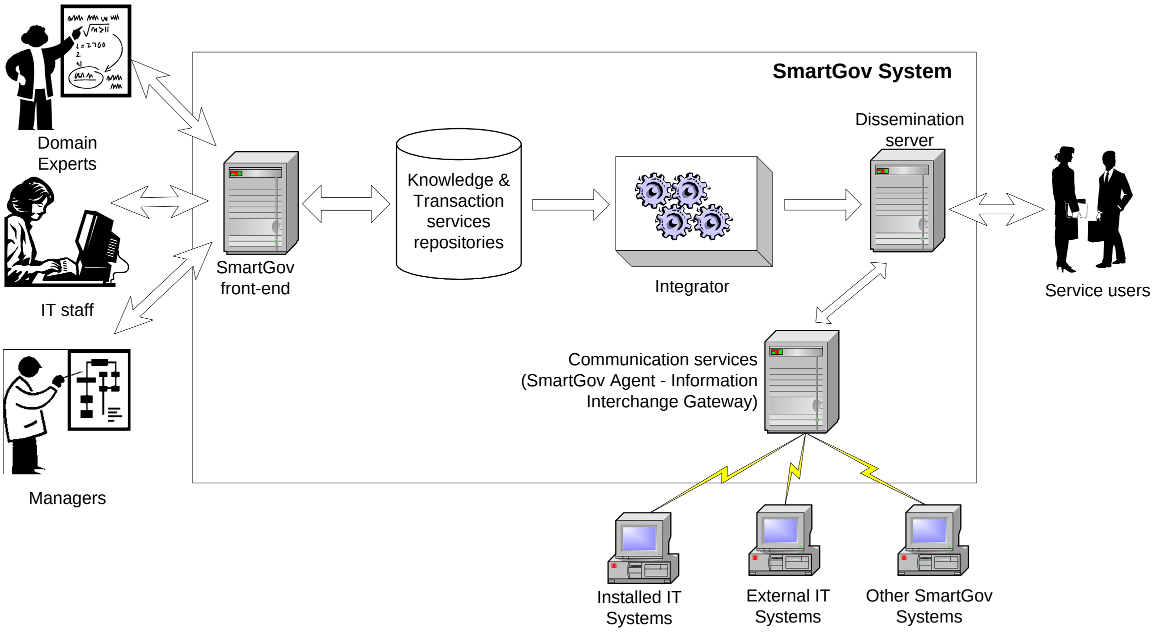

Figure 1: Architecture

This figure illustrates the SmartGov platform architecture, in which the following modules may be identified:

- The SmartGov knowledge and transaction services repositories. These are general depots for storing organisational knowledge and information pertaining to the transaction services that are developed using the SmartGov platform. In order to provide a semantically rich environment and facilitate extensibility and interoperability, all data is stored in XML format.
- The SmartGov front-end, which constitutes of personalised application development environments which are available to the actors involved in the lifecycle of electronic transaction services, namely domain experts, IT staff and managers. The actors employ the SmartGov front end to populate, query and modify the knowledge and transaction services repositories.
- The SmartGov integrator, a software module that reads the contents of the knowledge and transaction services repositories and automatically generates all

necessary elements (files, objects, components etc) for a fully operational transactional service. These elements are then deployed on a dissemination server to initiate delivery to users.

- The SmartGov communication component, comprising of two units, namely the SmartGov agent and the information interchange gateway. This module provides generic communication mechanisms with installed IT systems for the purposes of data exchange, hiding idiosyncrasies and peculiarities of information system platforms and facilitating resilience against temporary failures.

4. SmartGov knowledge and transaction services repositories

The development of the knowledge and the transaction services repositories constitute one of the major challenges of the SmartGov platform.

As a first step of the development, an analysis of the concept of transaction services was carried out. The results allowed us constructing abstract models for transaction public services. For analysis purposes, the extensible Markup Language (XML) was selected [6]. The use of XML ensures interoperability between SmartGov platforms but also between SmartGov platforms and external IT systems. It should be noted that the use of XML is recommended by a number of national initiatives e.g. the UK [7] and the Greek e-Government Interoperability Frameworks, the German SAGA etc. The use of XML is also widely used in e-business with the development of relevant standards, such as ebXML [8]. At the same time, a number of e-government initiatives, such as OASIS e-government working group are considering this technology. Finally, although the European Commission has not yet recommended a European Interoperability Framework (IDA is expected to provide that by the end of 2003 according to eEurope2005 action plan [9]) it is highly possible that this will also suggest the use of XML.

Within the SmartGov technical platform, four different XML document types have been specified, which cover all the information needed for the development of e-forms services:

1. The transaction service XML document: This document is the top-level element in the logical structure of the transaction service modelling. It contains information about the form documents (presented below) that constitute a service, validation rules at service level and other data pertinent to the service as a whole.
2. The Form XML document: Forms are the basic presentation and interaction unit for the end-user of the transaction service (e.g. citizens). In the context of the SmartGov platform, a form is divided into two parts:
   - The semantic part, which defines what information is entered in the form, the validation checks that apply to the form and the knowledge units (presented below), which will be presented to the user.
   - The layout part, which defines the appearance of the elements on the client device through which the electronic service is accessed (this document has an XHTML format).
3. The Transaction Service Element XML document: Transaction Service Elements (TSEs) are considered as the basic building block of an e-service. They are used to represent basic data types used within the organization. TSEs are not to be confused with basic data types as handled by programming languages. They are not just strings, integers, floats etc: TSEs are conceptual constructs that map onto the organization's practices. A TSE represents a real-world entity and its attributes model this entity's characteristics in a self-contained manner. TSEs are defined in an XML format and contain the following properties:
   - Unique identifier
   - Machine-oriented data type, e.g. integer, string, float etc.
   - Data type format rules

- Presentational info, possibly according to dissemination channel, e.g. length of data, number of decimals, colour, etc.
- Interface definitions for transforming the TSE values from and to different formats and for communicating with third-party systems.
- Generic name and/or service specific aliases (or handles)
- Generic validation constraints/conditions. Service specific constraints and/or more detailed ones are considered to belong to the Knowledge Repository.

TSEs are "cloned" when the time comes to implement a new service. The properties of the cloned TSEs can be overridden with service-specific properties. These properties are expected to be specialized versions of the properties offered by the generic TSE they clone, suitable for the service they refer to.

4. The Knowledge Unit (KU) XML document: Every entity within the platform may be associated with any number of knowledge units. These knowledge units may pertain to procedures in the platform (e.g. how a TSE is created), to specific items (e.g. for a TSE representing the passport number, an associated KU may describe the format of the passport number or contain the relevant legislation), or constitute help for the end users of the transaction service (e.g. KUs with descriptions, examples etc).

5. The use of the SmartGov technical platform

This section presents the use of the SmartGov platform in assisting public authorities to create and manage transaction services. For this purpose, we first present the phases in a transaction services lifecycle, according to an analysis that we performed:

1. The manager decides to launch a new service
2. The manager creates a working group, of domain experts, IT staff, managers and service workers.
3. The group creates the service requirements
4. The group creates the service specifications (process model)
5. The group develops the transaction service components
    - Forms (developed by domain experts and possibly IT staff)
    - KUs: Knowledge Units (developed by working group)
    - TSEs: Transaction Service Elements (developed by domain experts)
    - Validation checks (developed by IT staff and domain experts)
    - Links to back-end systems (developed by Domain experts and possibly IT staff)
    - Managerial statistics (developed by Managers and Domain experts)
    - IT-related statistics.
6. Integration of components
7. Testing
8. Evaluation
9. Deployment
10. Operation and maintenance
11. Feedback
12. Improvement
13. Discontinuation of a service

The SmartGov platform is mainly utilised in phases 5, 6 and 11. More specifically, in step 5 the users use the platform in order to describe all the necessary XML documents that together comprise the transaction services. The users use an intuitive and user-friendly interface in order to create and manage these documents, which are stored in the XML repository. The use of these tools do not require IT skills and actually an expression language has been designed and developed to enforce validation constraints.

In step 6, the integrator component of the SmartGov platform is employed. This component has a minimal interface and its task is to access the repository that holds the XML document describing a service, extract the required data and generate and deploy the web application. The generator interacts with several external modules like an XML parser (XERCES), an XSL Engine (XALAN [10]), Java-2-XML binding framework (Castor [11]) and build tool (ANT [12]) in order to achieve its goal. At the end it generates the appropriate modules required by the Apache STRUTS web application framework [13] and deploys the application in a web container.

Finally, in step 11 feedback is collected both by end users and via the statistics collection mechanisms of the SmartGov platform which will be exploited for service improvement in phase 12. In these phases, the SmartGov platform offers support for statistics collection, user account management and database backup and recovery.

6. Results and Business Benefits

The SmartGov platform provides the technological software infrastructure for public authorities to create and manage their online transaction public services. The results so far include specifying a technical architecture for the SmartGov platform. They further include specifying a number of XML documents to model transaction services. Currently, the SmartGov platform is under development. The next steps include finalising development as well as deploying and evaluating the platform in two public authorities, namely the city of Edinburgh and the Greek ministry of Economics.

The business benefit of the SmartGov platform is two-fold. Firstly, the SmartGov platform allows in-house development and management of transaction public services. This suggests a new approach that diverts from a common practice, which is to outsource the development and some times maintenance of such services.

Secondly, the SmartGov platform can be used by personnel without IT skills. The effort for creating and managing transaction services is therefore shifted from IT experts to domain experts and managers.

5. Conclusions

This paper presents the SmartGov platform, a knowledge-based technical platform to assist public sector employees to generate online transaction services by simplifying their development, maintenance and integration with installed IT systems.

The platform consists of four main components: the SmartGov knowledge and transaction services repositories, the SmartGov front-end, the SmartGov integrator, and the SmartGov communication component. The first of them contains all documents necessary for developing an online public service based on e-forms. To ensure interoperability these documents are stored in XML. More specifically, four different types of XML documents have been specified: The transaction service XML document, the Form XML document, the Transaction Service Element XML document, and the Knowledge Unit XML document.

The SmartGov platform allows public servants to handle a number of steps within a transaction service lifecycle in an intuitive, user-friendly manner.

Acknowledgments

The work reported in this paper was carried out within the project “A governmental knowledge based platform for public sector online services” (SmartGov) [2][3]. The SmartGov project commenced on 1st February 2002 and is co-funded by the European Commission under contract IST-2001-35399 of the 5th Framework Programme [14]. The partners of the SmartGov project are University of Athens, T-Nova Systems, Indra Sistemas S.A., Archetypon S.A., International Teledemocracy Centre (Nappier University), General Secretariat of Information Systems (Hellenic Ministry of Finance) and city of Edinburgh.

References

[1] European Commission, Public Sector Information: A Key Resource for Europe, Green paper on Public Sector Information in the Information Society, available at http://europa.eu.int/ISPO/docs/policy/docs/COM(98)585/.

[2] SmartGov project, http://www.smartgov-project.org

[3] E. Tambouris et al., SmartGov: A Governmental Knowledge-Based Platform For Public Sector Online Services. In: Proceeding of Knowledge Management in e-Government Workshop, 2002

[4] The Rational Unified Process, IBM Software Group, available at http://www.rational.com/products/rup/index.jsp

[5] T. Quatrani, Visual Modeling with Rational Rose 2000 and UML, Addison-Wesley, 2000.

[6] World Wide Web Consortium, The XML Specification, available at http://www.w3.org/xml

[7] UK e-Government Interoperability Framework (e-GIF) version 5, available at http://www.e-envoy.gov.uk/oee/oee.nsf/sections/frameworks-egif5/$file/egif5.htm

[8] ebXML, enabling a global electronic market, http://www.ebxml.org/

[9] European Commission, eEurope 2005: an information society for all, available at http://europa.eu.int/information_society/eeurope/news_library/eeurope2005/index_en.htm

[10] The Apache XML Project Xalan, available at http://xml.apache.org/xalan-j/

[11] The Exolab Group, The Castor Project, available at http://castor.exolab.org/index.html

[12] Apache foundation, The Apache Ant Project, available at http://ant.apache.org/

[13] The Jakarta project, The Struts Framework, available at http://jakarta.apache.org/struts/index.html

[14] European Commission, Administrations Home Page, available at http://www.cordis.lu/ist/ka1/administrations/home.html