



Technical University of Vienna
Information Systems Institute
Distributed Systems Group

Towards an Open Source e-Governance Solution

Benjamin A. Schmit, Nima Heschmat,
Schahram Dustdar, Gernot Müller and
Jeremy Solarz
benjamin@infosys.tuwien.ac.at
Nima.HESCHMAT@cec.eu.int
dustdar@infosys.tuwien.ac.at
gernot@muellersbuero.at
jeremy_solarz@utanet.at

TUV-1841-2006-39

May 17, 2006

Electronic Governance is becoming a more and more important topic in legislative bodies worldwide. In this paper, we present the architecture of an open toolsuite developed for the European Union project MetaLoGo, which aims to support Latin American municipalities with e-governance solutions. It has been developed using only open source components and is fully available under an open source license. The core components are a content management system providing the Web portal, a workflow management system handling administrative processes, and a customer relationship management system. The toolsuite can be deployed on outdated hardware as well and does not impose any license costs.

Keywords: e-governance, content management, workflow management

Towards an Open Source e-Governance Solution

Benjamin A. Schmit, Nima Heschmat, Schahram Dustdar,
Gernot Müller, Jeremy Solarz
Vienna University of Technology
Information Systems Institute
Distributed Systems Group
Vienna, Austria, Europe

Abstract

Electronic Governance is becoming a more and more important topic in legislative bodies worldwide. In this paper, we present the architecture of an open toolsuite developed for the European Union project MetaLoGo, which aims to support Latin American municipalities with e-governance solutions. It has been developed using only open source components and is fully available under an open source license. The core components are a content management system providing the Web portal, a workflow management system handling administrative processes, and a customer relationship management system. The toolsuite can be deployed on outdated hardware as well and does not impose any license costs.

1 Introduction

This paper describes the e-governance portal toolsuite that has been developed at the Vienna University of Technology in the context of the EU project MetaLoGo [6]. This project has been co-financed under the ALIS programme [13] of the European Union. It targets Latin American municipalities, which imposes a number of requirements on the software.

First, all software components used for the project had to be open source because of the license costs involved, and because this allows later modification in the destination environment. Another limitation was the lack of substantial resources available for the deployment of the software. Not all the 14 pilot municipalities have their own datacenter, nor can this be a requirement for the project. In some of them, a single PC connected to a slow modem provides the frontend of the municipality to the Internet.

This paper does not describe the social impact the project hopes to impose in the target municipalities, but instead focuses on the technical decisions and on the software architecture. The main components used were a content management system (CMS), a workflow management system (WFM), and a customer relationship management system (CRM). These components have been connected and enriched by additional components written at the Distributed Systems Group of the Vienna University of Technology.

Schahram Dustdar was responsible for the local project coordination at the group, while Benjamin A. Schmit coordinated the implementation efforts. Im-

plementation was done by Nima Heschmat, whose master’s thesis covered a part of the software, Gernot Müller and Jeremy Solarz.

The paper is structured as follows. Section 2 describes the overall system architecture. Section 3 presents related work that has been used for our software or was taken into consideration. The software components are described in the following sections: Content management (Section 4), workflow management (Section 5), and customer relationship (Section 6). Finally, the conclusion is drawn in Section 8.

2 Architecture

An overview on the architecture of our solution is shown in Figure 1. Sections 4–6 provide more details about the individual components.

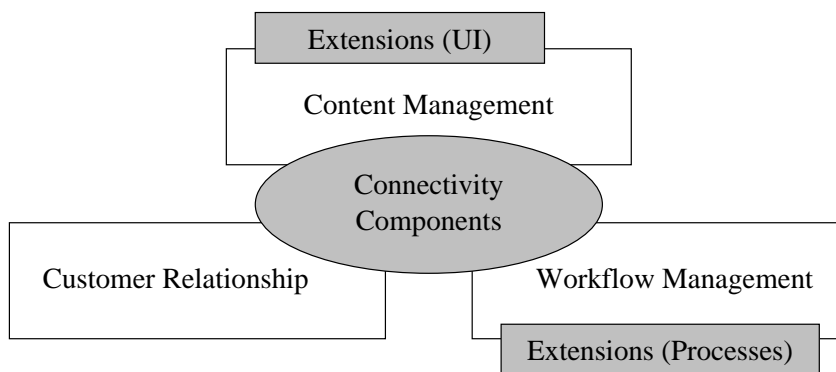


Figure 1: Architecture Overview

Web content is provided by a *content management system* (CMS). It also provides the complete user interface of the software, i.e. all functionality can be accessed online.

The CMS is divided into a frontend and a backend. The frontend is visible to all users of the Website, although parts of it may have restricted access. The backend, on the other hand, is available only to the officials of the municipality and for maintenance tasks.

Besides managing dynamic Web content, several frontend and backend extensions make it possible to access functionality provided by the other components. Since the CMS provides the software’s user interface, some of these extensions provide Web frontends to the other components. Additional CMS extensions by third party providers can be used together with our extensions.

The municipal processes are handled by a *workflow management system* (WFM) which provides support for basic workflows. Several extensions have been developed that handle advanced topics related to e-governance such as task assignment, user notification, and form handling. Process and form descriptions can be created or edited using a graphical editor.

Customer data is handled by a *customer relationship management system* (CRM). It stores data about both municipal officers (which can also be displayed on the Web frontend) and customers. CRM data is also used to provide

restricted frontend areas accessible only to customers, and for access control on process instantiation and execution.

Finally, some *Connectivity Components* link these core components together to create a single e-governance portal.

3 Related Work

ICT4D, the use of information and communications technologies for development co-operation, has gained value in its own right. Just like the water, power or transport sectors, it supports economic growth, as well as individual and community empowerment in developing countries [12]. The concept of an “information society” has been promoted by the European Union globally and programs like ALIS aim to advance development in this area in order to diminish the so-called “digital divide” [13].

ALIS encourages cooperation and innovation on various levels of citizen participation. While MetaLoGo is targeting small and medium enterprises (SMEs) to enhance business related processes, the eGOIA project [3] focuses on easily usable and accessible Citizen Access Points to enhance direct citizen participation. The effectiveness of e-governance initiatives in general, and the ALIS programme in particular, has to be considered in context of many related projects that are currently ongoing.

Specifically, the MetaLoGo toolsuite attempts to attain the previously mentioned goals by ensuring ownership through open source licensing and close collaboration with local partners. Moreover it seeks empowerment of local business communities by providing direct and transparent access to government transactions. The technical solution aims to accommodate these objectives at the different levels of implementation.

3.1 Content Management

For content management, we evaluated a number of content management systems for their suitability for the MetaLoGo project. A list of all open source CMS would be too long for this paper. Therefore, we reference two projects which provide structured information on a number of open source CMS [17, 2]. These projects were particularly helpful for the CMS selection task.

Three systems were taken into closer consideration: APC ActionApps [9], Mambo [11], and Typo3 [14].

ActionApps has already been used in Columbian municipalities and therefore support could have been provided easily and acceptance would have been high. However, it was found to be lacking several features that were necessary for the project, such as easy extensibility.

Finally, Typo3 was chosen as the most promising candidate, partly because of the large number of extensions available through the Typo3 Extension Repository. Re-use of already implemented extensions should keep the implementation efforts to a minimum.

3.2 Workflow Management

During the design phase various open source workflow management implementations were evaluated for the core engine, narrowing selection down to three options. ActiveBPEL [1] was considered for its BPEL standard support; however difficulties in developing feature extensions through web services and the lack of an open source license for its graphical designer lead to a decision against it.

OpenFlow [7], was considered due to its programming language PHP, facilitating simple integration in the context of MetaLoGo within Typo3, but rejected due to lack of solid developer support.

It was also attempted to provide the essential e-governance features through workflow systems directly integrated in various content management or customer relationship management systems. However simplicity and non-standard implementation of supported processes lead to the conclusion that an independent implementation must be considered.

This process lead to the development of OpenPELGO based on the JBoss jBPM engine [5].

For the implementation of OpenPELGO, jBPM has been selected as the workflow management engine due to its process standard support, solid implementation, and sizable open source developer community. Feature extensions to jBPM can be implemented with relative ease and the graphical process designer enables technical and governance partners to optimize their process design efforts.

E-Forms have been embedded into the system using FormFaces [4], an implementation of the XForms specification [10]. This open specification provides flexibility as well as standard compliance for designing Web forms.

3.3 Customer Relationship

After examining several open source customer relationship management tools (the most promising candidate was SugarCRM [19]), we decided that the integration of another large component into our system would result in more implementation work than the implementation of a small CRM itself. Therefore, it was decided to write a CRM on top of the Typo3 content management system, and to view data about municipal officers and customers as another type of content.

This decision removes the necessity of mapping the interface of the CRM to Typo3 — the user interface can be included in the CRM extension and parts of Typo3's backend user interface can also be reused. For this project, we also wouldn't have used large parts of existing CRM systems, which are mostly focused at customer support in a commercial environment.

4 Content Management

The Typo3 content management system [14] has been selected to provide the content management and user interface tool for our solution. In the future, it will also handle customer relationship management (see Section 6).

A major advantage of Typo3 is its extensibility. The system knows various types of extensions which enhance the server in numerous ways, by adding new

content types, library code or extending the frontend or backend view. Several such extensions have been developed in order to provide user interfaces to various e-governance tasks.

4.1 Process Deployment

In jBPM, workflows can be created with a graphical editor based on the Eclipse platform [15]. Support for process creation and editing in MetaLoGo will be described in Section 5.1.

The process deployment tool has been implemented as a backend extension to Typo3.¹ It is intended to be used only by system administrators when the e-governance toolsuite is first deployed or when a process has been changed.

The extension allows administrators to reset the database (this should only happen during process development, before the site is online), and to deploy and update process descriptions created in Eclipse. When a process has been selected for deployment or update, detailed information is shown, especially dependencies on other processes, if any.

Due to the jBPM implementation, existing processes that use an updated process description continue in their current form, while newly started processes use the new description. This feature allows smooth transitions between different process versions.

4.2 Workflow Initialization

The implementation of this small backend extension was necessary because the CRM tool had not yet been finished.² It allows the user currently logged into the Typo3 backend to start any of the deployed processes. A process is started by clicking on a link and will then proceed according to its process description.

4.3 Tasks and e-Forms

Process steps that are not handled automatically by the software need to be managed manually. Therefore, task assignment handlers have been written (Section 5.2.1) that assign a manual process task to an individual or a group.³

The backend extension allows a user logged into the Typo3 backend (usually a municipal officer) to see all processes currently waiting for user interaction that are assigned either to him or to one of the CRM groups the user belongs to. If a task belongs to a group, the user can accept handling of the process step.

As soon as a task has been assigned to a single user, this user may start editing attached form information. The form editing interface has been based on FormFaces [4], an implementation of XForms [18, 10]. Forms data may be stored without proceeding in the process, or the process step may be ended after filling out all required data. The entered form data may be used either at other manual steps of the process by other users, or also for automatic evaluations in the process.

¹This component has been implemented by Benjamin A. Schmit.

²The workflow initialization component has been implemented by Jeremy Solarz.

³The user interface for handling these steps has been written by Jeremy Solarz and Nima Heschmat.

Assignment of a process task to a customer has already been implemented within the process engine. The frontend extension, however, will be implemented through a restricted area within the CRM that has not yet been finished.

4.4 Monitoring

The process monitoring and statistics extension is currently still undergoing development.⁴ Statistics such as the mean run-time of a process with a given description or the number of processes that succeeded, failed, or returned errors have already been implemented. It is done by extracting data about past processes from the process database, and by following certain guidelines during process design (e.g. standardized names for end states that represent success or failure).

Monitoring and continuous evaluation is done by reusing the e-forms extension. For the evaluation of customer satisfaction, an evaluation form is attached to a process in addition to the standard process forms. It is shown to the customer after the process itself has finished.

Automatic evaluation of form data entered by the customers will be possible for radio or dropdown type form fields (percentage of customers selecting each option for text fields, mean value for number values) as well as for checkbox fields (percentage of customers that have selected an option). Textual feedback through these forms is also supported.

Internal evaluation (i.e. evaluation of feedback by municipal officials) is done in the same way. This time, the evaluation forms are not added to an existing process, but to a stand-alone process that can be started by any user who wishes to participate in the evaluation.

5 Workflow Management

The architecture of the workflow monitoring extensions, together with its user interface implemented as Typo3 extensions, is shown in Figure 2.

5.1 e-Forms

Municipal workflows are implemented in our solution as jBPM processes. The forms that are invariably associated with such workflows have been mapped to jBPM process variables. In order to facilitate form processing, editing, and permissions, they are stored as XForms data [18, 10], which can be easily updated during process execution.

Process design is handled through a graphical editor which comes with jBPM. This editor is based on the Eclipse integrated development platform [15]. It allows the creation of process graphs together with automatic or manual task nodes and decisions. Its output are JPDL documents, a process description language based on XML.

Within these process descriptions, forms can be attached using jBPM controller classes (see below). The forms can also be created in Eclipse using the

⁴It has been implemented by Benjamin A. Schmit.

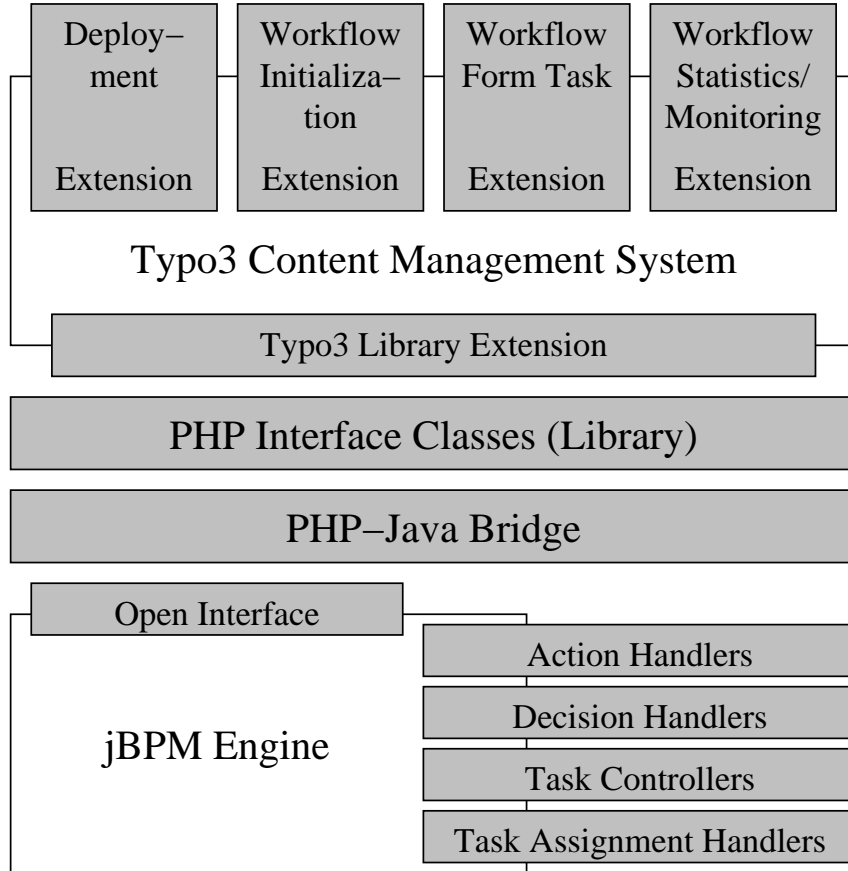


Figure 2: Workflow Management Overview

built-in graphical XML editor. Forms are XML documents which follow the syntax defined within an XML schema file (XSD). A form consists of one or more sections, which are logical units for form data access control within processes. Such a section consists of one or more form fields.

Form fields are defined by a name, a default value, possibly a number of options, and a field and a data type. The field type governs the appearance of the form field on the screen, e.g. a text field. The data type, which may be assigned independently from the field type, selects a storage type (e.g. integer).

5.2 OpenPELGO

OpenPELGO, based on the JBoss jBPM engine and programmed in Java, aims to provide process management functions required in the context of electronic governance (e-governance) transactions. The acronym stands for “Open Pro-

cessing of ELeCtronic GOVernance”.⁵

The architecture of OpenPELGO is structured in three layers. jBPM, configured to utilize MySQL as its database in MetaLoGo, functions as the core workflow engine and makes up the foundation, on top of which the specific features related to the e-governance context are programmed. On the top level the OpenPelgoInterface implements persistence calls to enable extended process operations and provides an interface aimed at extending the features to content management systems and other applications, which serve as a user interface.

5.2.1 Controllers and Handlers

The specific e-governance features are designed by extending and customizing respective jBPM controller and handler classes, including implementations for process handling, electronic form handling, and G2B communication.

Customized assignment handlers are designed to delegate process tasks to respective personnel or administrative units. Newly developed task controllers integrate document and form handling into jBPMs task structure.

The document extension, based on the XForms standard, offers multiple form support per task and stores corresponding XML files within jBPM as process variables. Flexibility of the XForms standard guarantees, that information delivery can be assured across various platforms and even to non-conventional devices.

Process or form relevant user data can be stored or retrieved across the CMS or CRM interface. Process delegation is controlled by various action handlers capable of reassigning tasks or redirecting the process based on predefined decisions, as well as logical tests on process document data or form input.

The communication module, also triggered by action handlers, assures that information, especially documents and forms, can be transmitted to the respective parties by various means (e.g. email, fax, SMS).

Finally, an exception handling scheme is designed to assure smooth execution of the flow and correct delegation in cases of process failures.

5.2.2 The OpenPelgoInterface

The above mentioned extended features to jBPM can be accessed through the OpenPelgoInterface. It has been designed with content management systems in mind, but can also be accessed by other applications.

The interface implements functions for standardized e-governance operations, handles data and document flow between the interacting components and utilizes jBPM’s persistence feature for extended processes. OpenPELGO also includes interfaces to content management and customer relationship management databases for user management and verification purposes.

5.3 PHP and Typo3 Interface Classes

In order to access the OpenPelgoInterface (written in Java) from PHP code, the PHP-Java-Bridge [8] has been included within the software solution. It allows

⁵This component has been developed by Nima Heschmat as the central topic of his master’s thesis. Further development efforts for this software are now coordinated by a SourceForge project [16].

PHP code to instantiate Java classes and to call member functions and use attributes through PHP wrapper objects.

On top of the bridge, a PHP interface layer has been implemented that maps the OpenPelgoInterface to PHP. It also maps the result values and possible exceptions thrown by the Java code into PHP wrapper objects. Above this layer, a Typo3 library extension incorporates this functionality into the CMS, where it is used by various other Typo3 extensions (Section 4).⁶

6 Customer Relationship Management

The CRM component is currently still future work. It is intended to cover both the display of contact data for officials in charge through the CMS frontend and the interaction of customers with municipal processes in a restricted frontend area.

The main data types are companies (departments), users, and roles. For an officer, the municipal department he has been assigned to is stored in the company record, together with a flag designating him as an internal user. The roles assigned to him manage access control to process steps defined in the WFM component. For a customer, the role designates the permissions he has within the company, e.g. whether he may sign contracts for the company.

A Typo3 extension should display the logical organization of the municipality and allow the listing of all officers on a department website. Backend users with appropriate permissions may use the extension to browse through customer data or modify it.

A second Typo3 extension will handle process interactions with customers. A process which allows registration for the restricted frontend area can be started by all website users. After the process terminates successfully (which usually includes some authentication checks), the customer gets a password for his account. Then, process steps assigned to this customer as well as e-forms assigned to these steps can be handled in this area by the customer. Also, new processes can be started there according to the roles that have been assigned.

7 Evaluation

Because the implementation is still ongoing, full evaluation is not yet possible. The potential for improvements in the partner municipalities, however, is enormous: The process for obtaining a business operation license in the municipality of Comayagua, Honduras, before the implementation of our system took between two months and two years. With the process redesign and automation, the duration could be shortened to a reliable two weeks.

A very important aspect for an e-governance system is security. This aspect has been handled by reusing the infrastructure of the Apache HTTP server and the Typo3 CMS. All sensitive data can be transmitted through secure channels, and a flag in the CMS allows the administrator to specify which pages must not be transmitted without encryption.

⁶The PHP interface has been designed by Gernot Müller, the Typo3 library extension by Jeremy Solarz.

Authentication has also been implemented in Typo3. Per default, only hash values of passwords are transmitted via the Internet, so that the password itself is reasonably safe even without encryption. To guard against imposter or replay attacks too, encryption has to be enforced again in the CMS.

All data used by the system is stored in a database. Per default, MySQL is used here, but other database servers can be used as well. Since even partially performed process steps are stored persistently, the software allows the level of reliability needed by government authorities even in the unlikely event of a complete software crash. System reliability can therefore be improved by adding hardware resources, such as redundant storage for the database server. If such resources cannot be acquired by the municipality, additional reliability is possible by adding hardcopy tasks at some milestones in the municipal processes.

8 Conclusion

Our work unifies a content management system, a workflow management tool, a customer relationship management system, and various smaller tools into a single e-governance platform that is completely available under open source licenses. Technically, the main advantages of this solution are the single Web user interface available through Typo3, the incorporation of the sophisticated process management engine jBPM together with support for e-form management following the XForms specification, the possibility for future extensions through the open source license, and the low resource requirements.

The toolsuite has been developed with Latin American municipalities in mind, but can be deployed with equal efficiency on all kind of e-governance portals. Furthermore, the interface to the workflow engine, OpenPelgoInterface, can be reused on other content management systems as well. This provides the possibility to use workflow management also in other website contexts and on other types of portal servers.

We hope that this solution will prove to be a valuable contribution to the open source community even beyond the scope of the targeted municipalities and the MetaLoGo project.

References

- [1] ActiveBPEL website. <http://www.activebpel.org/>.
- [2] The CMS matrix website. <http://www.cmsmatrix.org/>.
- [3] eGOIA website. <http://www.egoia.info/>.
- [4] FormFaces website. <http://www.formfaces.com/>.
- [5] jBPM website. <http://www.jboss.com/products/jbpm>.
- [6] MetaLoGo website. <http://www.metalogo.org/>.
- [7] OpenFlow website. <http://www.openflow.it/>.
- [8] PHP-Java-Bridge website. <http://php-java-bridge.sourceforge.net/>.

- [9] APC. ActionApps website. <http://actionapps.org/>.
- [10] John M. Boyer, David Landwehr, Roland Merrick, T.V. Raman, Micah Dubinko, and Leigh L. Klotz. *Xforms 1.0* (second edition). W3C Recommendation, March 2006.
- [11] Robert Deutz. *Mambo: Installation, Administration, Anwendung und Entwicklung*. Springer, September 2005.
- [12] European Commission ICD Thematic Network. *Guidelines on ICTs and Development*. February 2005. Prepared by ict Development Associates ltd. Draft.
- [13] European Union. Website of the ALIS programme.
http://europa.eu.int/comm/europeaid/projects/alis/index_en.htm.
- [14] Rene Fritz, Daniel Hinderink, and Werner Altmann. *Typo3: Enterprise Content Management*. Packt, June 2005.
- [15] Erich Gamma and Kent Beck. *Contributing to Eclipse: Principles, Patterns, and Plugins*. Addison-Wesley, October 2003.
- [16] Nima Heschmat. OpenPELGO website.
<http://sourceforge.net/projects/pelgo/>.
- [17] OSCOM. CMS matrix website. <http://www.oscom.org/matrix/>.
- [18] T. V. Raman. *XForms: XML Powered Web Forms*. Addison-Wesley, September 2003.
- [19] Michael J. R. Whitehead. *Implementing SugarCRM*. Packt, January 2006.