Strategies for Virtual Enterprises using XForms and the Semantic Web

Albert Rainer, Jürgen Dorn, Peter Hrastnik

Electronic Commerce Competence Center (ec3)
Donau-City-Straße 1
1220 Wien Austria
albert.rainer@ec3.at
juergen.dorn@ec3.at
peter.hrastnik@ec3.at

Abstract: Virtual enterprises provide customized services to parties. Business processes of such a virtual enterprise often use Web forms for data manipulation and user navigation. We describe how forms originating from different organizations of a virtual enterprise can be integrated into a single application by deploying XForms, a standard for user interfaces defined by W3C. Based on a scenario that considers the organization of a scientific conference as a virtual enterprise, we show different strategies which can be applied to overcome the problems of heterogeneous data formats and heterogeneous user interfaces using Semantic Web technologies and XForms.

1 Introduction

ec3 (www.ec3.at) is a research center funded by different industrial, governmental and academic organizations. With m>Forum a project was started that brings together competence from different industrial partners (a telecom partner, an ERP partner, as well as partners from conference organization and tourism) and different basic research projects (projects on user interface design, data and web mining, cluster analysis, business modeling and evaluation, mobile services and virtual enterprises) of ec3.

The project shall support the organization of scientific conferences with innovative solutions in the area of multi-modal communication (i.e. mobile and Internet communication as well as local communication techniques such as Bluetooth or radio frequency technology) and personalization of services for customers.

ec3’s Move-project aims at a development framework to support virtual enterprises [1]. A virtual enterprise (VE) is a geographically dispersed and semi-stable network of independent organizations that depends on innovations and is based on core competencies as well as mass-customizations [5].
A major challenge for a VE is its ability to plug arbitrary services from arbitrary organizations into the existing organization and to provide a low entry barrier for potentially new members. Thus in the Move-project we develop tools to configure business process at runtime and to achieve customized services to customer requests. Business processes of such a VE often use Web based forms for data manipulation and user navigation. We a framework that addresses the needs and requirements of form-based business processes and explore the deployment of XForms and Semantic Web techniques in the following.

1.1 XForms

XForms is a W3C specification for generation of Web based forms that promise improvements compared to "normal" HTML forms in terms of authoring, reuse, localization, accessibility, usability, and device independence. XForms is conceptually composed of three major parts that provide separation of data, presentation, and logic. These parts are Model, UI Controls, and the Processing Model.
- A Model section can contain several child elements such as instance, bind, and submit. The instance element has a dual role. First, it serves as a template for XML data that will be submitted to the server and second, it provides initial data to populate the XForms.
- XForms defines an abstract set of UI controls. Each control specifies its binding to the XML instance data via either an XPath expression or the id of a bind element. Typical UI controls such as input, textarea, or radio buttons are provided.
- The Processing Model has a set of event handlers that respond to user interaction events and perform state transitions. Events can be of type Initialization, Interaction, Notification or Error Conditions.

1.2 Semantic Web

The Semantic Web as defined by W3C is based on RDF and provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries [7]. Since XForms is based on XML and instance documents can reside somewhere on the Internet, RDF is the natural candidate for the additional functionality needed to integrate disparate UIs into a single application. An additional reason why we prefer RDF to other knowledge representations, for example KIF, is that up-to-date tools and APIs are on-hand.

1.3 Use-cases

Applications built with XForms and additionally extended with semantic information have a number of possible use cases.
- Quality assurance: The XForms standard itself defines means to maintain and improve quality of applications. In addition, Semantic Web techniques help to further increase this quality.
The application can maintain a similar look and feel. In the best case, users are not aware of the different service providers but see the VE as an unity. The application can support internationalization for help texts, labels, and other relevant information.

Members of the VE that provide Web accessible services can describe these services and provide information about their forms, how they are used, what they provide, and the requirements they expect.

- Reuse of existing forms: A well designed form can be considered as a valuable asset that may be reused by other applications or even by other members of the VE. To accomplish forms reuse, means for introspection and configuration have to be provided. Additionally, it helps form reusing when localization techniques are applied, i.e., several forms share the same instance sources that can easily be maintained and adapted at a single site. With these means it is possible to compile forms into aggregates, as for example a wizard that contains forms from several members of the VE.

- Reasoning: Beyond simple if-then-else declarations reasoning offers the opportunity to customize the application at runtime in order to deliver the desired business value for the parties.

- Model checking: XForms can be considered as lightweight application having events, conditions, and actions. Formal methods like model checking offer means to assert certain requirements, for instance, to assure that credit card payment takes place before an applicant for a conference receives a formal registration document.

- Planning: If a party wishes to have a certain service provided, it should be possible to derive an optimal sequence of required forms automatically.

1.4 Related Work

The problem of heterogeneous data in inter-organizational business processes is addressed for example in [3][2], where mapping of disparate documents is discussed. The frameworks MAFRA [6] and RDFT [4] provide tools that help in defining mappings or so-called semantic bridges between a source and a target ontology. On the other hand, standardization attempts like xCBL[1] or UBL[2] try to define an universal ontology for business documents and document exchange. Usage of such standards would greatly simplify data exchange between parties, but praxis shows that numerous formats coexist. Finally, some industrial efforts (e.g. Novell)[3] exist to build applications from XForms documents using a graphical designer to develop forms and the navigation structure, but data integration is not considered.

---

3 Novell Xforms, www.novell.com/xforms
2 Strategies

In order to use XForms to integrate user interfaces from different VE members having potentially different data formats a strategy has to be embarked to handle the resulting problems. We will present a scenario and show the different strategies a VE can apply.

2.1 Scenario

We consider a scientific conference as a VE. Different partners such as a scientific organization as for example IEEE, a local organization, resource providers (e.g. a hotel) and general service providers such as a telecommunication company or a bank collaborate for the time of the conference to supply the conference participants with optimal services [1]. Partners will provide different forms to participants. Traditionally, a participant must fill out Web forms for the local organizer, the accommodation provider, the credit card organization and others. Web based forms are used by various stakeholders to enter new data, edit existing ones, make agreements, manage resources, administer user access rights, and other services. For example, a person who is a member of an associated organization may get a discount when applying for a conference. In the application form he/she can select these organizations and subsequently the forms compiled by these organizations are presented and the verification procedure is performed. Next in the process, a form coming from a partner company that provides financial services is presented and the clearing process takes place.

2.2 Problem Description

When building an application from heterogeneous sources, i.e. XForms belonging to different VE members, the data format will be heterogeneous, too. While one organization uses \( ec3:mail \) to denote the email input field and responds to the membership question with \( ec3:member="yes\|no" \), another organization may have \( ocg:email \) and responds \( ocg:membership=\"verified\|unverified\" \). A solution to this problem might be to force all parties to comply to a single ontology. A second approach is to have the data objects mapped to each other and store these mappings in an external mapping document. Both approaches have assets and drawbacks. While an overall data format provides an easy way of integration at the process level, i.e. the engine that performs the actual processing of the various forms, the VE members have to agree to a certain format and streamline their backend application to handle this global format. Since an XForms User Interface may play a role in other applications too (e.g. the credit card service provider has a single form to enter credit card information and process it), forms have to be maintained for each VE unless all VEs consent to this very same format. On the other hand, when allowing disparate formats the entry barriers for an prospective member of the VE are lower since no adaption of backend systems is needed and no duplications of forms have to be maintained. In contrast, the VE has now to maintain the mapping documents and the processing engine has to perform translations at runtime which could lead to performance problems.
For our prototype we have chosen the second approach, i.e. member organizations having their own data formats and provide mappings and perform data transformations during business process execution. Based on the sample process from the scenario we will show how Semantic Web techniques can help to ease these problems.

2.3 RDF/XML-based Data

Data instances used in XForms are simply well-structured XML documents. Since RDF and related Semantic Web techniques (RDFS, OWL) have an XML representation it is possible to have RDF instance models as a source for XForms instances. The advantages of using RDF syntax in XForms instances are:

- it puts us in a position where mappings can be defined in terms of OWL constructs (for example owl:sameAs or owl:equivalentProperty relations).
- it allows us to use technology already on-hand that provides RDF serialization and reasoning support.

A drawback of this method is that service providers have to adapt their existing instance data format, although the adaptation is quite simple. The adaptation in detail works as depicted in Figure 1. The XML instance is surrounded with an RDF block, sequences of XML elements are denoted as RDF collections.

```xml
  <ec3:Conference acronym="AEIOU">
    <ec3:startDate>XX-XX-XXXX</ec3:startDate>
    <ec3:Associates rdf:parseType="Collection">
      <ec3:Org name="OCG"/>
      <ec3:Org name="SAP"/>
    </ec3:Associates>
  </ec3:Conference>
</rdf:RDF>
```

Figure 1: XML instance document adapted for RDF processing

2.4 Enriching XForms Elements with Semantic Information

Having the instance data template allows us to reason about the data but gives no hint about the form that works on these data. For example, an instance document may contain numerous elements while the form has access to only a few elements. In order to provide means for forms introspection and increase accessibility (e.g. for non-visual browsers or software agents) additional information has to be provided. Semantic Web techniques can be used again to perform this task.

The XForms specification declares an extension element that can be used inside an arbitrary XForms element to add information not directly related to the form itself. This element can contain RDF that provides information about it. The example in Figure 2 shows an input text field with an additional extension element declaring this input field having an email address as value.
With this knowledge, an agent can infer that this form has a parameter $e0$ that expects an email as value and act accordingly, for example, generate an http request. An advantage of this kind of entering additional descriptions is that the extension element can be placed directly as a child to any XForms element thus gluing the RDF tightly to the relevant element. A drawback is that forms developers are not necessarily skilled in RDF or similar Semantic Web languages.

3 Implementation

We have implemented our framework as a prototype to study several aspects of XForms combined with Semantic Web techniques. The architecture is built at the server-side with the XForms engine Chiba (chiba.sourceforge.net) running in a Tomcat servlet container (jakarta.apache.org/tomcat), the Jena API (www.hpl.hp.com/semweb) for RDF processing and Jess (herzberg.ca.sandia.gov/jess) as a reasoning engine. At a high-level view it works as follows on our simple scenario: When the process starts (i.e., the first form is loaded into the XForms engine) it instantiates an RDF/XML data container that is used throughout the process. The container can contain default values (e.g., associated organizations) or empty declarations (the user data, membership to associate) that are provided by the user or other applications. When the next form is loaded (e.g., the form from the associate that verifies membership), mapping from the container to the form takes place (the user data are propagated). In the next step, the result from the membership verification request is mapped into the container.
4 Conclusion

Our current prototype framework shows that virtual enterprises can perform Web based business processes using XForms applications. Moreover, integration of disparate data using Semantic Web techniques provides low entry barriers for potential partners. But there are still some open issues. One direction is to investigate the possibility to extract additional information from the Processing Model part of XForms documents and generate an OWL-S (OWL-based Web service ontology) description that captures not only the data and control elements but also the process and data flow. Another question to answer is performance and scalability of our approach in a “real world” application.

5 References