

# eFormsDirect – XML-basiertes eGovernment-Framework für Intelligente Formulare auf der Basis von XForms

Stefan Audersch, Thomas Courvoisier, Guntram Flach

Zentrum für Graphische Datenverarbeitung e.V., Rostock  
Joachim-Jungius-Str. 11, 18059 Rostock  
{audersch, tc, gf}@rostock.zgdv.de

**Abstract:** Mit Hilfe innovativer XML-Technologie können Anforderungen neuartiger, verteilter Anwendungen und der damit in Zusammenhang stehende Informationsbedarf im Rahmen eines universellen, skalierbaren Multimedia Content Managements befriedigt werden. Eine zunehmende Bedeutung kommt in diesem Zusammenhang der Einbeziehung einer adäquaten XML-Middleware sowie einer intelligenten Interaktions- und Dialogsteuerung zu. Mit dem XForms-Standard liegt eine Spezifikation vor, auf deren Grundlage die flexible, plattform- und endgeräteunabhängige Repräsentation von Online-Formularen in XML möglich ist. Ziel dieses Beitrages ist die Konzeptionierung einer offenen und erweiterbaren XForms-Frameworkarchitektur unter Nutzung der Web-Service-Technologie als Kommunikations- und Koordinierungsdienst. Signifikante Erweiterungsmöglichkeiten ergeben sich dabei durch die Kopplung der Web-Service-Interfaces mit RDF-Content-Repository-Ansätzen im Sinne semantisch angereicherter Web-Services. Die Schwerpunkte verteilte XML-basierte Request-Verarbeitung und personalisierte Interaktions- und Dialogsteuerung werden speziell anhand des eGovernment-Anwendungsfeldes intelligente Formulare (eFormsDirect) vorgestellt.

## 1. Einleitung

eGovernment verändert die Struktur und Kultur von öffentlichen Verwaltungen. Zum einen geschieht dies durch interne Reformprozesse im Rahmen der Einführung von Workflow-Systemen zur elektronischen Abwicklung von Geschäftsprozessen. Zum anderen präsentieren öffentliche Verwaltungen fast täglich neue Serviceleistungen via Internet, die eine neue Qualität von Bürgerservice mit sich bringen. Im Verhältnis von öffentlicher Verwaltung zum Bürger und Wirtschaft überwiegen derzeit die Informationsdienstleistungen, dicht gefolgt von Kommunikationsdiensten. Aber der eigentliche Kern von eGovernment besteht aus Transaktionsdiensten, und dabei befinden wir uns noch in der Anfangsphase.

Im Rahmen der Pilotanwendung „Bürgerkontaktsystem Mecklenburg-Vorpommern“<sup>1</sup> (BKS-MV) wird zurzeit eine Lösung für eine eGovernment-Plattform entwickelt, die

---

<sup>1</sup> Im Rahmen der Multimedia-Initiative des Landes Mecklenburg-Vorpommern arbeiten die Deutsche Telekom (T-Systems Nova Berlin), die Universität Rostock, das Fraunhofer IGD, die Planet AG und das ZGDV Rostock gemeinsam mit dem DVZ M-V GmbH an der Entwicklung des Bürgerkontaktsystems Mecklenburg-Vorpommern (BKS).

relevante Dienste-Angebote der Verwaltung an den Bürger und die Wirtschaft abbilden kann. Ziel ist es, im Land Mecklenburg-Vorpommern eine neue Generation von eGovernment-Angeboten zu etablieren, die den gewachsenen Bedürfnissen der Bürger Rechnung tragen, Behördenvorgänge unter Nutzung verschiedenartiger Medienzugänge zu unterstützen. Ein Teil bei den Umsetzungsarbeiten von BKS ist die Entwicklung einer Technologie zur Bereitstellung von Online-Formularen. Die hierfür entwickelten Ansätze und Realisierungsarbeiten werden in diesem Beitrag näher dargestellt. Anforderungen an das System sind unter anderem die Nutzung offener Standards, Integrationsmöglichkeiten für elektronische Bezahlungsverfahren, elektronische Signatur etc. und ein geringfügigster Installationsaufwand für den Bürger bei der Nutzung der Formulare. Neben diesen sind natürlich Anforderungen an die Sicherheit und den Schutz persönlicher Daten, die Zwangsläufe

Bei den Realisierungsarbeiten wird auf verfügbare Standards (u.a. W3C, OSCI) aufgesetzt, um die Offenheit, Erweiterbarkeit und Plattformunabhängigkeit der BKS-System-Dienste zu gewährleisten. OSCI<sup>2</sup> (Online Service Computer Interface) ist ein speziell für die synchrone und asynchrone Nachrichtenübermittlung entwickelter Protokollstandard (u.a. OSCI-Transport 1.2), welcher die sichere Abwicklung von Transaktionen zwischen Bürgern und Verwaltung erlaubt. Diese Sicherheit bezieht sich sowohl auf die Möglichkeit der digitalen Signatur der transportierten Inhalte als auch auf die Vertraulichkeit. Zudem soll durch eine Strukturierung der Nachrichteninhalte (basiert auf XML-Schema) die automatisierte Weiterverarbeitung ermöglicht werden. Die Entwicklung von OSCI orientiert sich an den einschlägigen Standards und Vorgaben, insbesondere dem Signaturgesetz und der Signaturverordnung sowie XML.

Beim Entwurf der grundlegenden OSCI-Funktionalitäten wurde im Rahmen des Anwendungsszenarios am Anfang eine Orientierung auf den Ersatz des konventionell unterschriebenen Papierformulars durch ein digital signiertes, „elektronisches Formular“ vorgenommen. Nach diesem Leitbild würde ein signiertes Java Applet durch Benutzerinteraktion Inhaltsdaten generieren, die dann vom Bürger elektronisch signiert und als OSCI-Nachricht in Richtung Verwaltung gesandt werden. Nach unbestimmter Zeit erhält der Bürger als Reaktion einen Bescheid der Behörde. Diese Sichtweise orientiert sich am klassischen Formular- und Papierweg in der Kommunikation zwischen Bürger und Verwaltung. Zukünftig wird es hier allerdings eine Schwerpunktverschiebung in Richtung unmittelbarem, synchronem Zugriff mittels OSCI (direkte Anbindung eines Fachverfahrens) geben. Die gegenwärtigen Entwicklungsarbeiten orientieren sich an dieser Anforderung im Rahmen der Anwendung „Melderegisterauskunft“ unter Berücksichtigung weiterer relevanter Standards und Schnittstellen-Spezifikationen (u.a. OSCI-Xmeld 1.1<sup>3</sup>, OK-EWO).

Im Nachfolgenden wird ein kurzer Überblick über die durch das ZGDV Rostock entwickelte Framework-Architektur mit server- und client-seitigen Komponenten gegeben. Ausgehend von einer XForms-basierten dynamischen Generierung der Online-

---

<sup>2</sup> OSCI ist ein MEDIA@Komm-Ergebnis. Der Standardisierungs-Prozess wird derzeit durch die OSCI-Leitstelle Bremen im Auftrag der KoopA-ADV (Kooperationsausschuss ADV Bund/Länder/kommunaler Bereich) fortgeführt.

<sup>3</sup> „OSCI-XMeld 1.1“ wurde speziell für die Anforderungen des Meldewesens (einfache Melderegisterauskunft, Rückmeldung, Fortschreibung des Melderegisters) entworfen. Ergebnis ist ein XML-basiertes Informationsmodell für die Objekte des Meldewesens, das mit dem bundeseinheitlichen Teil des DSMeld abgeglichen ist.

Formulare wird im Rahmen der prototypischen Implementation „eFormsDirect“<sup>4</sup> die Ableitung von Unterstützungsmöglichkeiten für die obige Aufgabenstellung innerhalb des projektspezifischen BKS-Anwendungsszenarios „Intelligente Formulare“ untersucht. Die Middleware-orientierten Kommunikationsprozesse wurden auf der Basis von Web-Service-Schnittstellen umgesetzt.

## 2. Systemarchitektur

Der in diesem Abschnitt vorgestellte Ansatz ist Teil der Realisierungsarbeiten für das Bürgerkontaktsystem MV und ermöglicht die einfache, flexible Bereitstellung sowie Nutzung von Online-Formularen im Bereich eGovernment. Zentrale Schwerpunkte bei der Entwicklung waren sowohl Aspekte der Sicherheit, Modularität sowie die Interoperabilität.

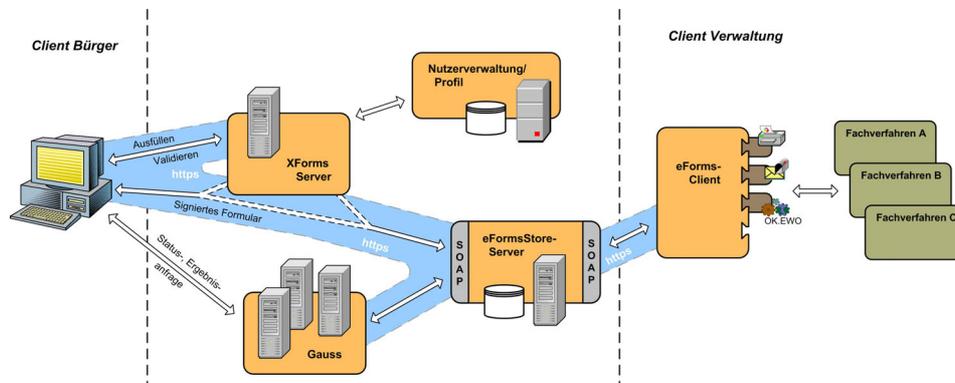


Abbildung 1: eFormsDirect Architektur

Das System unterteilt sich in drei Teile, wobei sich der mittlere Teil aus mehreren Serverkomponenten zusammensetzt:

### Client für den Endanwender (Bürger)

Der Endanwender benötigt zur Nutzung der Online-Formulare einen einfachen Browser. Dieser stellt die vom XForms-Server bereitgestellten Formulare dar und ermöglicht das Ausfüllen. Ist eine Signatur und Verschlüsselung<sup>5</sup> des Dokumentes notwendig, so erfolgt dieses durch ein Java-Applet in Kommunikation mit dem XForms-Server. Die Signierung der Dokumente beruht auf dem PKCS#7-Standard [PK93], wobei clientseitig eine TCOS2.0- bzw. Netkey2000-Chipkarte und ein entsprechender Kartenleser zum Einsatz kommt.

### XForms Server, eFormsStore-Server, Nutzerverwaltung, Gauß

<sup>4</sup> Im Rahmen des eFormsDirect-Frameworks werden derzeit weitere funktionale Komponenten umgesetzt (u.a. eFormsAuthor, eFormsSecure, eFormsSignature, eFormsPay), auf die im Rahmen dieses Beitrages nicht eingegangen werden kann.

<sup>5</sup> Verschlüsselungsverfahren: RSA/ECB/PKCS1PADDING

Zentraler Kern der Architektur sind der XForms Server (siehe Abschnitt 3.1) und der eFormsStore-Server (siehe Abschnitt 3.2). Der XForms Server stellt auf der Basis von Formular-Definitionen verschiedene Formulare zur Nutzung bereit, validiert die Formulare beim Ausfüllvorgang und übernimmt beim Einstellen der Dokumente Teile der Signatur und Verschlüsselung. Aufgabe des eFormsStore-Servers ist die Haltung von ausgefüllten Formularen und Dokumenten für den jeweiligen Empfänger.

Willigt der Endbenutzer einer erweiterten Nutzung von Online-Formularen zu, bei der personenbezogene Daten serverseitig gespeichert werden, so können mit Hilfe der Nutzerverwaltung Formulare vorausgefüllt werden. Intelligente Softwareagenten sammeln in diesem Fall die dafür notwendigen Informationen. Gauss<sup>6</sup> als Content-Management-System stellt die zentrale Komponente der Pilotanwendung BKS-MV dar. Über das System erfolgt unter anderem die Präsentation der Verwaltungsobjekte, der Einstieg des Bürgers zu den Online-Formularen oder auch der Zugriff auf Status und Ergebnis von eingestellten Formularen. Die beiden Komponenten Nutzerverwaltung und Gauss werden in diesem Beitrag nicht weiter betrachtet.

#### **eFormsClient (Verwaltung)**

Der Zugriff (im Anwendungsfeld eGovernment von der Verwaltungsseite aus) auf die eingestellten Dokumente ist mit dem eFormsClient möglich. Der Client kommuniziert direkt mit dem eFormsStore-Server über SOAP. So ist eine Kommunikation auch im Falle einer zwischengeschalteten Firewall möglich. Wie der Bürger, so identifiziert sich auch die Verwaltung mit Hilfe einer Signatur. Die für den entsprechenden Empfänger relevanten Dokumente werden anhand des Zertifikates vom eFormStore-Server ermittelt und mit dem privaten Schlüssel zur Bearbeitung entschlüsselt. Der eFormsClient verfügt zur manuellen bzw. elektronischen Weiterverarbeitung der Dokumente über verschiedene Schnittstellen (z.B. PDF, Druck, eMail, OK.EWO<sup>7</sup>).

Durch die Dreiteilung der Architektur erfolgt die Kommunikation zwischen Client des Bürgers und eFormsClient nie direkt, sondern stets asynchron über den eFormsStore-Server, bei dem die Dokumente verschlüsselt und nur mit einer lesbaren Information über den Empfänger vorliegen.

Im folgenden Kapitel werden die wichtigsten Komponenten (XForms Server und eFormsStore-Server) und die für deren Entwicklung grundlegenden Ansätze näher vorgestellt.

### **3. Realisierungsaspekte**

#### **3.1 XForms Server**

XForms ist eine Spezifikation von Web-Formularen, die auf einer Reihe von Plattformen, wie PCs, Handhelds oder anderen informationsverarbeitenden Geräten verwendet werden kann. XForms wird von der XForms Working Group beim W3C

---

<sup>6</sup> Gauss: CMS der Firma Gauss Interprise AG

<sup>7</sup> OK.EWO: AKDB-Verfahren zur elektronischen Verarbeitung von melderechtsrelevanten Vorgängen in Einwohnerverwaltungen

spezifiziert. Die Spezifikation hat gegenwärtig den Stand eines Last Call Working Draft und hat damit einen bereits recht stabilen Zustand erreicht [DK+02].

Wie in anderen modernen Web-Applikationen besteht auch bei XForms der Ansatz in der Trennung von Zweck und Präsentation. So wird der „Zweck“ eines Formulars in einem XForms Model beschrieben. Dieses Model enthält ein Datenmodell, das die Formulardatenstruktur definiert und die Beschreibung von Ereignissen (z.B. durch Nutzereingaben), auf die das Formular reagieren muss. Dieses XForms Model kann mit einer Reihe von Nutzeroberflächen (z.B. XHTML, SVG) zusammenarbeiten.

Die Zahl der Implementierungen von XForms ist noch überschaubar. Zum einen ist zunächst der X-Smiles- Browser zu nennen, bei dem es sich um einen experimentellen Web-Browser handelt, der die Möglichkeiten der neuesten W3C Standards demonstriert. Auch eine erste XForms-Spezifikation wurde umgesetzt [HV02]. Mit dem X-Smiles Browser existiert ein XForms-fähiger Client, der ein XForms Model laden und interpretieren kann und damit der XForms-Spezifikation entspricht.

Im vorgestellten Anwendungsszenario ist allerdings davon auszugehen, dass der Nutzer als Client lediglich einen gebräuchlichen Web-Browser einsetzen kann bzw. einzusetzen bereit ist. Damit war eine XForms-Implementierung zu wählen, die entweder die clientseitige Funktionalität über ein Browser-Plugin, wie z.B. eine Java Virtual Machine, oder über Javascript-Elemente realisiert, oder aber eine Implementierung, die serverseitig die Formularlogik umsetzt. Alle Ansätze haben ihre Vor- und Nachteile. Im Rahmen des Projektes wurde der serverseitige Ansatz gewählt, der größtmögliche Client- bzw. Browserunabhängigkeit gewährleistet, bei gleichzeitigen Vorteilen in der Erstellung und Pflege komplexer Formularanwendungen.

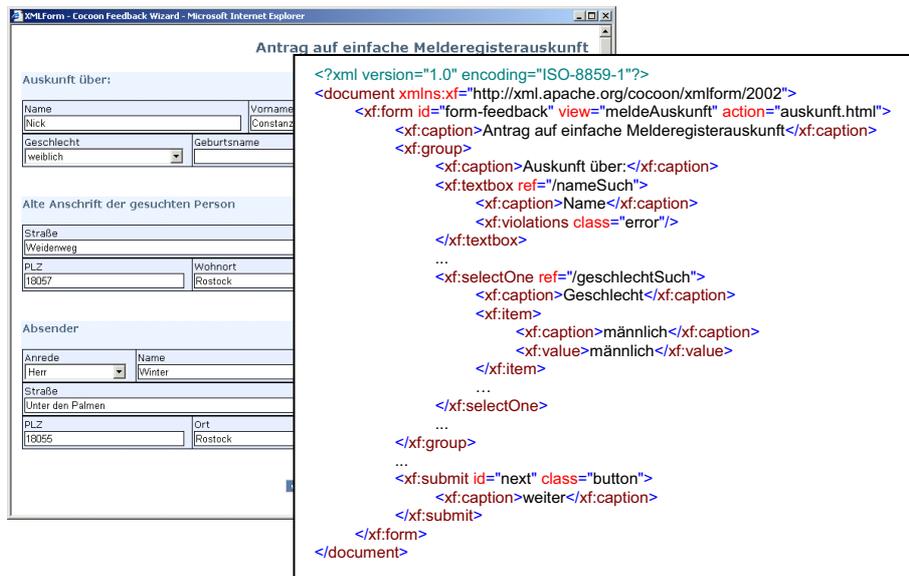
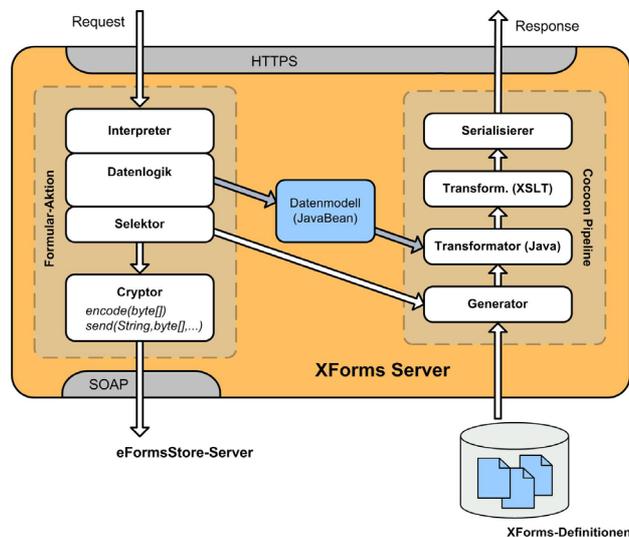


Abbildung 2: Abstrakte Formularbeschreibung in XMLForms

Als Grundlage der Implementierung wurde die XMLForm-Komponente [Co02] von Cocoon gewählt, die eine Untermenge der XForms-Spezifikation umsetzt, aber komplett auf clientseitige Ereignisse und Aktionen verzichtet. Die XMLForm-Komponente ist ab Version 2.1 in Cocoon integriert. Es handelt sich bei dieser Formularimplementierung um eine Umsetzung der Prinzipien von XForms. Es ist keine konkrete Implementierung der Spezifikation selbst. Der Ansatz von XMLForms bietet eine strikte Umsetzung des MVC-Paradigmas, wobei die Präsentation der Formulare gemäß dem Cocoon-Ansatz über Stylesheets realisiert wird. Abbildung 2 zeigt ein Beispiel einer abstrakten Formularbeschreibung gemäß XForms-Syntax und das mittels Stylesheets umgesetzte Präsentationsergebnis in HTML. Das Form-Model wird über eine JavaBean beschrieben und enthält sämtliche Definitionen, die die Datenstruktur im Formular betreffen.

Den Steuerungsteil eines Formulars bildet die Implementierung einer Cocoon-Aktion, die auf bestimmte Zustände und Ereignisse im Formular reagiert. Optional existiert ein Datenschema, das zur Integritätsprüfung und Validierung der Nutzereingaben verwendet werden kann. Gegenüber herkömmlichen Formularen bietet die Implementierung auf der Basis von XMLForm die Steuerung komplexer Formularlogik und damit die Bewältigung einfacher wie komplexer Aufgaben durch den Nutzer bei relativer Unabhängigkeit vom verwendeten Client. Das objektbasierte Datenmodell ermöglicht die Strukturierung der Formulardaten und damit sowohl die Erfassung komplexer Datenmengen als auch die effizientere Verarbeitung. Durch die unabhängige Präsentationskomponente ist eine Konsumierung der Formulare auf Clients wie Voice-Browsern und Handheld mit relativ geringem Aufwand umzusetzen, dies gilt insbesondere dadurch, dass auf clientseitige Ereignisse und Aktionen verzichtet wird.



**Abbildung 3:** XForms Server

Basierend auf der XMLForm-Komponente von Cocoon wurden im Rahmen des Projektes eine Reihe von abstrakten Klassen und Schnittstellen definiert, die zum einen die Implementierung komplexer Formularlogik vereinfacht und zum anderen die

Grundlage für einen Formulareditor (eFormsAuthor) bilden sollen, der die Erstellung von Formularen im eGovernment-Bereich auch auf einem solch komplexen Level auf einfache Weise ermöglichen soll. Abbildung 3 zeigt den schematischen Aufbau des XForms Servers.

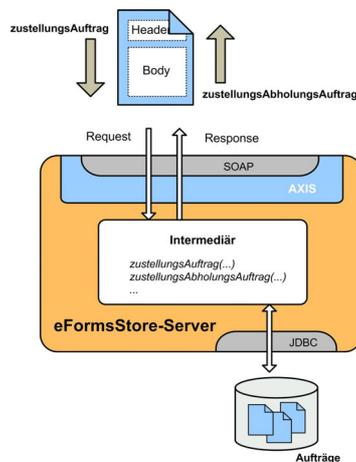
### 3.2 Web-Service-Interfaces

Angelehnt an den OSCI-Standard wurde zur Kommunikation zwischen Bürger und Behörden ein Intermediär, der eFormsStore-Server entwickelt. Dieser ermöglicht den indirekten Austausch von Daten zwischen Sender und Empfänger nach dem Prinzip *One-Way-Message, aktiver Empfänger*. Der eFormsStore-Server stellt dabei ein Webservice dar, basierend auf dem Transportprotokoll SOAP (Simple Object Access Protocol). Ein Ausschnitt der Beschreibung des Webservices findet sich in Listing 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<wsdl:definitions ...>
+ <wsdl:types>
  <wsdl:message name="zustellungsAuftragRequest">
    <wsdl:part name="content1" type="xsd:base64Binary"/>
    <wsdl:part name="messageId" type="xsd:string"/>
    <wsdl:part name="subject" type="xsd:string"/>
    <wsdl:part name="certificateAddressee1" type="xsd:base64Binary"/>
  </wsdl:message>
  <wsdl:message name="zustellungsAbholungsAuftragRequest">
    <wsdl:part name="certificateAddressee" type="xsd:string"/>
    <wsdl:part name="messageId" type="xsd:string"/>
  </wsdl:message>
  <wsdl:message name="zustellungsAbholungsAuftragResponse">
    <wsdl:part name="return" type="xsd:string"/>
  </wsdl:message>
+ <wsdl:portType name="EFormsDirectStoreServer">
+ <wsdl:binding name="EFormsDirectStoreServerSoapBinding" type="intf:EFormsDirectStoreServer">
+ <wsdl:service name="EFormsDirectStoreServerService">
</wsdl:definitions>
```

**Listing 1:** WSDL-Beschreibung für den eFormsStore-Server

Wesentliche Funktionen des Servers sind Methoden zum Einstellen und Abholen von Aufträgen (Formulare, Antworten, Dokumente, etc.). Daneben existieren beispielsweise noch Funktionen zur Statusabfrage. Ein Auftrag wird in einem Container verpackt. Dieser besteht aus einem Header und einem Body. Der Header beinhaltet allgemeine Informationen über den Auftrag und das Zertifikat des Empfängers. Der Body ist mit dem öffentlichen Schlüssel des Empfängers verschlüsselt und enthält die eigentlichen Auftragsdaten sowie das Zertifikat des Senders. Beim Intermediär erfolgt die temporäre Speicherung der Aufträge bis zu deren Abholung in einer Relationalen Datenbank.



**Abbildung 4:** eFormsStore-Server

Signifikante Erweiterungsmöglichkeiten ergeben sich im Rahmen der Weiterentwicklung in Richtung einer universellen Frameworkarchitektur z.B. durch die Kopplung der Web-Service-Interfaces mit RDF-Content-Repository-Ansätzen [LS99] im Sinne semantisch angereicherter Web-Services [FBM02]. Zu den Kernpunkten der Arbeiten, die sich derzeit in der Anfangsphase befinden, gehören insbesondere die Bereitstellung universeller Schnittstellen und Prozesse zwischen RDF-Repositories, Mediator-Komponenten, domänenspezifische Ontologien sowie Web Services. Innerhalb eines derartigen Ansatzes können so intelligente Web Services geschaffen werden, die semantisch gesteuert z.B. die Integrations- und Transformationsprozesse im Sinne dynamischer Services realisieren können.

Im Projekt XPEA (XML-basierte Plattform für wireless-orientierte E-Business- und Content-Management-Applikationen) [DCF01, FG00] des ZGDV Rostock wurden in diesem Zusammenhang erste Komponenten und Werkzeuge zur semiautomatischen semantischen Beschreibung von Datenbeständen entwickelt [Au02, ACF02, Co01]. Ziel des Projektes war die applikations- und medienunabhängige Integration heterogener Datenquellen sowie deren Adaption in einem mobilen Anwendungsumfeld. Die Anpassung von Daten und Datenquellen auf verschiedene Applikationen, Geräte und Medien war effektiv nur durch eine semantische Erfassung der Dateninhalte zu gewährleisten. Als Mittel zur semantischen Beschreibung wurde RDF verwendet.

#### **4. Prototyp-Realisierung eFormsDirect**

Die hier vorgestellten Realisierungsansätze wurden prototypisch im System eFormsDirect umgesetzt. Die Implementierung der einzelnen Module basiert, außer wenige Shared Libraries für die Signatur, vollständig auf Java. Für den Betrieb des Systems sind folgende APIs bzw. Technologien notwendig: Tomcat, Axis, Cocoon, ein beliebiges RDBMS. Die Kommunikation zwischen den einzelnen Komponenten erfolgt mit XML.



- [ACF02] S. Audersch, T. Courvoisier, G. Flach: Universeller Zugriff auf multimediale Dokumentstrukturen auf der Basis von RDF und MPEG-7, In Tagungsband GI-Workshop XML Technologien für das Semantic Web (XSW), Berlin, 2002.
- [Co01] T. Courvoisier: RDF-basiertes Content-Repository im Rahmen eines wireless-orientierten Multimedia-Informationssystems, In Tagungsband: 13. GI-Workshop Grundlagen von Datenbanken, Gommern, 2001.
- [Co02] Apache Cocoon Project: XMLForm Handling, <http://xml.apache.org/cocoon/userdocs/concepts/xmlform.html>, 2002.
- [DCF01] C. Dittberner, T. Courvoisier, G. Flach: Using Object-relational Database Systems and XML in the Context of an wireless-oriented Multimedia Information System, In Proc. of the 2th Int. Conference on Web-Age Information Managment, (WAIM), Xian, China, Juli 2001.
- [DK+02] M. Dubinko, L. Klotz, R. Merrick, T.V. Raman: XForms 1.0, W3C Working Draft, August 2002.
- [FBM02] D. Fensel, C. Bussler, A. Maedche: Semantic Web Enabled Web Services, In Proc. Of the First Int. Semantic Web Conference (ISWC), Italien, 2002.
- [FG00] G. Flach, G. Günther: Architecture for the Interaction and Access on Multimedia Database Systems in the Context of Mobile Environments, In Proc. of the 4<sup>th</sup> Int. Database Engineering and Application Symposium (IDEAS), Yokohama, Japan, 2000.
- [HV02] M. Honkala, P. Vuorimaa: Advanced UI Features in XForms, In Proc. of the 8th Int. Conference on Distributed Multimedia Systems, September, 2002.
- [LS99] O. Lassila, R. Swick: Resource Description Framework (RDF) Model and Syntax Specification., W3C Recommendation, 1999.
- [PK93] PKCS#7: Cryptographic Message Syntax Standard. An RSA Laboratories Technical Note, Version 1.5, Revised November 1, 1993.