OPEN XML COURT INTERFACE

Electronic Filing Manager Development Plan

April 26, 2004

counterclaim

MTG
Management
Consultants

TABLE OF CONTENTS

Page

I. DEVELOPMENT PLAN

I. DEVELOPMENT PLAN

This document describes the development plan for the Open XML Court Interface (OXCI) Electronic Filing Manager (EFM). The EFM will be developed in three iterations. Each iteration will include unit testing and integration testing. At the end of Iterations 1 and 2, the source code, documentation, and testing reports will be released to the pilot sites and other interested parties for early evaluation and testing. At the end of Iteration 3, the source code, documentation, and testing reports will be released for pilot testing.

A. ITERATION 1

Iteration 1 will include the following function points:

- Schema Compliance and Validation
- Court Policy
- Court Filing
- Generic Case Management System (CMS) and Document Management System (DMS) Adapter

Each of these function points is detailed below.

1. Schema Compliance and Validation

Create SchemaLxmlValidator.

Currently, the OpenEFM contains an interface for validation against the LegalXML Document Type Definition (DTD). An implementation that validates against the OXCI Court Filing schema will be developed.

Create a test case to validate sample filing with SchemaLxmlValidator.

A JUnit test case will be written to validate the sample OXCI Court Filing XML document against the OXCI Court Filing schema.

2. Court Policy

Add a Court Policy module to the EFM.

Currently, Court Policy is not supported. The code base will be modified to include this.

Create a Court Policy interface.

The Court Policy interface will define the methods used to access all information contained in the OXCI Court Policy schema. This information includes where to retrieve the Court Policy file, where to possibly host the file, and the data contained in the file.

Create an OXCI Court Policy implementation.

The OXCI Court Policy implementation will validate any XML document against the OXCI Court Policy specification. It will also contain the specific XPath queries for data retrieval that are unique to the OXCI specification.

Create a test case verifying proper loading of module data during configuration.

This basic test will ensure that all combinations of module attribute entries in the EFM XML configuration file are loaded correctly into the Court Policy object.

Create a test case verifying retrieval of a Court Policy XML document.

This test case will ensure that an XML Court Policy document can be retrieved from a URL, as well as from the local file system.

Create a test case verifying local hosting of a Court Policy XML document.

This test case will ensure that the EFM is capable of hosting the XML Court Policy document in its local Web service container.

3. Court Filing

Accept an OXCI Court Filing XML document via SOAP with ebXML payloads and test case.

Currently, a Simple Object Access Protocol (SOAP) filing interface is provided; however, filings are expected to validate against a DTD and no ebXML SOAP elements are used. The module that receives filings will be reimplemented to validate against the OXCI Court Filing schemas and to store specific data in ebXML SOAP elements. A test case that ensures the functionality of this transaction will also be constructed.

Query an OXCI Court Filing XML document for information via XPath.

Data will be retrieved from the XML Court Filing document by using XPath queries. These queries will be specific to the structure of the OXCI Court Filing schemas.

Create a test case for XPath queries.

A JUnit test will be provided to ensure that all XPath queries are executed successfully.

4. Generic CMS and DMS Adapter

Provide pseudo code for a generic CMS adapter.

The CMS adapter element is court-specific. An interface is already provided. A review of this interface will be conducted to determine if any further functionality is needed to support CMS and DMS interfaces. A generic implementation will be provided. This implementation will contain pseudo code for a generic approach to accepting a legal filing.

B. ITERATION 2

Iteration 2 will include the following function points:

- Query and Response
- J2EE Compliance
- Enforcement of Court Policy Data
- Update of Installation Guide
- Creation of ebXML Integration Instructions

Each of these function points is outlined below.

1. Query and Response

A Query and Response module is in place but only exists as an empty interface. An object representing the XML Queries and Responses will be defined and created. A transceiver capable of receiving the XML Queries and Responses will be defined and created. Application logic capable of sending XML Queries and Responses will be defined and created. OXCI-specific implementations of the needed interfaces will be created. Unit tests will be developed to test this functionality.

The OXCI EFM will act as middleware between an Electronic Filing Service Provider (EFSP) and a CMS. The general accepted flow of queries and responses involves travel through the EFM. EFSPs will create XML Queries. These queries will be passed on to the OXCI EFM through the Query Transceiver. Queries that the OXCI EFM has no ability to respond to (i.e., everything except the Policy Query) will be passed on the CMS system. When an XML Response is received from the CMS, it will be passed on to the requesting EFSP. Queries will be implemented in a synchronous fashion.

2. J2EE Compliance

During this time, the EFM code base will be evaluated and modified for compliance with the J2EE 1.0. Compliance will be determined by the ability to launch the application under a J2EE Web application component. Once compliance with the J2EE environment has been established, the OXCI EFM will be capable of running in two different modes. The first mode is the current manner in which the application runs. The current mode is running as a Web application, contained in a Jetty Web Server environment. This allows for an easy stand-alone execution of the application. The second mode will be running as a deployed application in any J2EE-compliant application server. Creation of a Web application Enterprise Archive (EAR) file will be available through the Ant build environment. The EAR file that is produced will be capable of being deployed in a J2EE environment.

3. Enforcement of Court Policy Data

The data made available to the EFM via the Court Policy module will be integrated into the application control flow. Test cases will validate that required data elements are enforced. Specifically, this includes operational requirements such as those outlined by the Rules Type. These rules are:

- Accept Document URL.
- Accept Documents Requiring Fee.
- Accept Sealed Documents.
- Accept Multiple Filings.
- Legal Envelope Maximum Size.

Each filing entering the system will be checked against these indicators to ensure that it complies with the Court Policy. Filings not in compliance with these rules will not be accepted into the OXCI EFM. They will instead be rejected with the reason clearly stated.

4. Update of Installation Guide

The OpenEFM project that the OXCI EFM is based on includes several different types of documentation. As a result of the many changes required by the OXCI EFM, much of the documentation has become outdated. The Installation Guide is one such document. The Installation Guide will be brought up to date during the second development iteration in order to allow for pilot sites to use the guide during the third development iteration.

5. Creation of ebXML Integration Instructions

OXCI EFM receives filings via a SOAP ebXML interface. In order to file against this interface, a Filing Service Provider will need to use an ebXML implementation to construct a SOAP message with certain data elements stored in certain SOAP fields. These details will be discussed in an ebXML Integration Instructions document. This document will be made available to any pilot sites or EFSPs that wish to integrate with the OXCI EFM via ebXML.

C. ITERATION 3

Iteration 3 will include the following function points:

- Payments
- CMS and DMS Connectors and Adapters
- EJB 2.0 Compliance

Each of these function points is outlined below.

1. Payments

The payment processing functionality of the EFM will be tested. JUnit tests will be created to ensure functional working order according to the OXCI Payment schema. UBL will be used to receive billing information.

Payments will be implemented in the following way: OXCI Payment schema will be included in the ebXML filing envelope. The OXCI Payment schema is based on types defined by UBL. The aforementioned schema will deliver the payment information between the EFSP and the EFM. The EFM will utilize preexisting payment processing logic to facilitate the actual transaction. These transactions will be conducted through VeriSign Inc.'s Payflow Pro system.

2. CMS and DMS Connectors and Adapters

Adapters for interfacing with the CMS and DMS at pilot sites are scheduled for development by counterclaim in the third iteration. These adapters will be working implementations of the generic CMS and DMS adapters for interfacing with the CMS and DMS at other pilot sites. In some cases the pilot sites may develop these adapters, and counterclaim will provide any needed support.

The CMS and DMS adapters will also be made available as EJB session beans. This will allow for these modules to be easily incorporated into other J2EE environments.

The CMS and DMS connectors will be accompanied by a SOAP implementation. This implementation will work with a provided Web Services Description Language (WSDL) to define the filing connection process.

3. EJB 2.0 Compliance

The objects defined in the model package (objects representing courts, EFSPs, users, filings, etc.) will be implemented as EJB 2.0 entity beans. This will allow the object persistence to be handled cleanly in a J2EE application server. Other objects that represent business logic, such as the CMS and DMS connectors and adapters, will be implemented as EJB session beans when appropriate.

II. DEVELOPMENT SCHEDULE

II. DEVELOPMENT SCHEDULE

The development schedule is part of the OXCI Project Work Plan document. Iteration 1 has been completed. Currently, the project is in development for Iteration 2. The schedule for the three iterations and pilot testing is as follows:

Iteration	Start Date	Completion Date
Iteration 1	February 17, 2004	March 23, 2004
Iteration 2	March 23, 2004	April 23, 2004
Iteration 3	April 23, 2004	May 21, 2004
Pilot Testing	May 21, 2004	July 12, 2004

For more detail on the overall project schedule, please consult the OXCI Project Work Plan.