# Project Document Cover Sheet

## *FINAL REPORT*

## Project

| | | | |
|---|---|---|---|
| **Project Acronym** | ONIXLT1 | **Project ID** | |
| **Project Title** | Development of an ONIX-PL licence expression drafting system | | |
| **Start Date** | 1/11/2006 | **End Date** | 30/3/2007 |
| **Lead Institution** | Book Industry Communication (BIC) | | |
| **Project Director** | Brian Green | | |
| **Project Manager & contact details** | Brian Green, BIC, brian@bic.org.uk<br>020 7607 0021 | | |
| **Partner Institutions** | | | |
| **Project Web URL** | | | |
| **Programme Name (and number)** | PALS 2 | | |
| **Programme Manager** | Christine Baldwin | | |

## Document

| | | | |
|---|---|---|---|
| **Document Title** | Final Report | | |
| **Reporting Period** | | | |
| **Author(s) & project role** | Brian Green, BIC, Project Manager | | |
| **Date** | | **Filename** | JISC_OPLE_finalreport_V3.doc |
| **URL** | | | |
| **Access** | ☐ Project and JISC internal | | ☐ General dissemination |

## Document History

| Version | Date | Comments |
|---|---|---|
| 1 | 31/5/07 | |
| 2 | 8/6/07 | |
| 3 | 12/6/07 | |

JISC
**PALS 2**

## Acknowledgements

The project is part funded as an extension of the JISC PALS2 Metadata project call. Further funding has been provided by the Publishers Licensing Society and EDItEUR.

## Executive Summary

This project has developed a fully functional prototype, based on the functional requirements for tools for creating ONIX-PL expressions of licences specified as part of the *JISC PALS 2 Electronic Expression of Licensing Terms: Specifying publisher tools and library benefits* project. We have called the ONIX-PL editing tool OPLE.

OPLE allows both licensors and licensees to create ONIX-PL expressions of their licenses and to revise drafts in the negotiation process. The system is based on the use of a library of templates that can be used to create precise expressions of individual licenses.

OPLE has been designed on a web architecture. Web browsers are used to provide a user-friendly graphical user interface and two views of the license expression are supported. These are a form view for creating the license expression and a page view for review, comparison with original licenses and printout.

It had already been agreed that the drafting system would be implemented on a popular web-based client-server technology platform in order to keep the requirement for bespoke software development to a minimum and make it easy to find the resources for developing and maintaining the bespoke software components. The system employs standard web browser clients to provide user interfaces to the system. This will make it relatively easy to roll out access to the system to any number of users, thereby supporting adoption of the system by both smaller and larger organisations.

The decision to base OPLE on free/open source software (FOSS) components has resulted in some delays and initial performance problems as the software on which OPLE is based was still undergoing frequent revision. However, the decision was inevitable given the cost restraints and the need to make the software freely available, and the initial problems were eventually overcome. A more detailed account of the technical development issues is appended as Appendix 2.

The following are two typical user scenarios. Others will no doubt be identified in the light of experience.

In the first, the user will interact with the ONIX-PL annotated schema that is pre-installed in OPLE to generate one or more templates based on their own boilerplate licences, from which they or other colleagues can derive specific licence expressions (i.e. expressions of individual licence agreements with particular customers or suppliers). This scenario requires some understanding of ONIX-PL in order to produce the initial template, and may require expert assistance at that stage; but once a template has been set up, the derivation of individual licence expressions should be straightforward.

In the second scenario, the user would select a generic licence template, probably based on a model licence, from the ONIX for Licensing Terms section of the EDItEUR website, and use this as a much simpler starting point to create their own template, which would then be used to derive specific licence expressions in the same way as in the first scenario. In addition to the JISC licences, it is EDItEUR's intention to produce further generic templates based on model licences.

The first users of OPLE are JISC Collections who have recently installed the system on their server and will be using it to map all their publisher licences to ONIX-PL.

Although the major international library systems vendors producing ERM (Electronic Resource management) systems have undertaken to implement ONIX-PL, none have yet done so. A tender for an ERM system issued by JISC Collection includes this requirement and there are indications that this is speeding up implementation by library systems vendors.
.

## Background

This project was an extension to a previous JISC PALS2 metadata project, JISC PALS 2 *Electronic Expression of Licensing Terms: Specifying publisher tools and library benefits*. This project produced *Functional requirements and design specification for an ONIX-PL license expression drafting system.* A detailed functional review of the OPLE Version 1 ONIX-PL license expression drafting system, against the original requirements and design specification, is included as Appendix 1. Further information about the ONIX-PL format for the expression of licensing terms can be found on the EDItEUR website www.editeur.org.

## Aims and Objectives

The aim of the project was to develop a fully functional prototype, based on the functional requirements for tools for creating ONIX-PL expressions of licences specified as part of the JISC PALS 2 *Electronic Expression of Licensing Terms: Specifying publisher tools and library benefits* project. These tools should be open source and be freely available to any potential users and should, therefore, themselves be based on open source components.

## Methodology

The project team decided to develop the software using X-Forms, a relatively recent W3C standard for handling XML-based forms on the web. The system has been implemented, as originally specified, as a server-side web application.

The server-side technology consists of a number of FOSS (Free/Open Source Software) components based on the Java platform, which provides a useful degree of insulation from the underlying hardware and operating system users will have. The system is already installed and running on both the Windows and Linux operating systems, and the user interface has been proved to run satisfactorily on three major browser technologies (Microsoft Internet Explorer, Mozilla Firefox and Opera).

The design is such that users need only have network access to the server, and a web browser client, to enjoy the full functionality of the software. It is also possible for the server and client to be run on a single PC avoiding the need for a separate server.

The user interface is under control of the open standard XForms specification, which allows a rich user interface to be constructed, offering more than legacy web-browser forms, and giving the application a “look and feel” closer to that of a traditional desktop application.

The system architecture is such that most processing takes place on the server. All interactions with clients are with XML data conveyed via the HTTP protocol, and client-side processing is limited to Javascript for providing a rich user interface and dynamic page updating. This technology mix is often referred to as AJAX -- Asynchronous JavaScript and XML.

The engine we are using, "Orbeon Forms" (www.orbeon.com) provides an application framework for the functionality specified for the system. This includes:

- Interaction with underlying XML data (ONIX data) via a rich user interface
- Transformation and processing of ONIX data with technologies such as XSLT
- Storage and querying of collections of XML data using the embedded eXist database.

## Implementation

OPLE has already been installed on the JISC Collections server and a staff member is being trained to use the tools to map JISC collections licences to ONIX-PL. It is hoped that this, combined with a JISC tender for an ERM system that can ingest and act on licences expressed in ONIX-PL format, will speed up the implementation of ONIX-PL in library systems and that publishers will then use OPLE to map their licences to ONIX-PL. There has also been interest shown by a major publishing software provider in building on the OPLE tools.

## Outputs and Results

The main project deliverable is a fully functional prototype of OPLE, an ONIX-PL license expression drafting system based on the previously defined specification, that can be used by publishers and libraries to draft ONIX-PL license expressions that can be communicated between licensee and licensor. The software, when sufficiently stable, will be available at http://ople.sourceforge.net, from which information about the technical architecture of OPLE and installation guidance are already available. A current pre-release version is available from the ONIX for Licensing Terms section of the EDItEUR website http://www.editeur.org

The report on development of the tools, *OPLE technical development issues*, appended to this report should be of interest to others building tools based on open source components.

An early design decision was to build an ONIX-PL editing tool that uses a client-server architecture based upon free open source software (FOSS) components. The result is an open source tool that is freely available to the whole Internet developer community to adapt and develop further, should they so wish, under terms and conditions that are entirely typical of such projects and assert the key rights of EDItEUR.

Quite late in the project, when we started to test the tool with realistic data (JISC NESLi2 model license), it became apparent that, although the tool software was proving in many respects remarkably robust, there were some serious performance problems with the tool as implemented. While the original implementation worked well for simple data structures, performance deteriorated quite rapidly as the data structures and the dynamic forms required to edit these structures became more complicated. We had not anticipated that the complexity of the ONIX-PL schema and the first realistic test data would make such heavy demands on memory and processing power that the tool was functional but in practice unusable due to the length of time taken to perform routine functions. As a consequence we

had to re-design, re-code and re-test some core functions at a late stage in the project. This meant that significant time was lost and this was the direct cause both of delays in completion of the project and of the omission from Version 1 of functionality that would otherwise have been included.

For these reasons the following functionality that has been omitted from Version 1, and which will be included in the next version of the tool to be released at no further cost to JISC

- the ability to update license expression templates and instances when the ONIX-PL schema is revised
- the ability to import sets of clauses from one template into another

Some functionality, which is desirable but not essential for the use of the tool, has not been implemented because the complexity of implementation has proved too great for a short development project, including:

- change tracking in revision of license expression templates and instances (data model implemented in Version 1, but functionality to use it not yet implemented)
- highlighting differences between distinct versions of a template
- automated tool updates, including automated downloading of schema revisions

Some functionality has not been implemented because the precise business requirement needs further specification in the light of user experience with Version 1 of the tool, including:

- the definition of user 'levels' in addition to "user" and "manager"
- validation of license expression templates and instances against specific business rules not expressed fully in the ONIX-PL schema
- export of license expressions in formats other than ONIX-PL and HTML

Additional functionality that has was not originally specified but which is to be added to a future version of the tool includes:

- the ability to create a read-only archive copy of a template, to avoid it being inadvertently edited.

## Outcomes

An immediate outcome of this project is the ability for JISC Collections to use OPLE to map all their individual licences into the ONIX-PL format and make them available to libraries in a standard machine-readable form. The tools will also enable both publishers and libraries to map existing and future licences into the standard ONIX-PL format.

Looking forward, EDItEUR will setup an OPLE users listserv for current and potential users to provide feedback and suggest improvements and additional functionality to OPLE.

Sourceforge will be a valuable forum for discussion of a more technical nature concerning the technical design and implementation of OPLE.

## Conclusions

The development process was much more complex than anticipated, especially given the problems with the open source components used. However, we see no practical alternative to the FOSS approach for this sort of project and continue to believe that this was the correct implementation approach

## Implications

JISC Collections will use OPLE to map their existing publisher licences to the ONIX-PL format. These will be made available to libraries either directly into their own systems or on a JISC repository. The availability of these licences will lead libraries to request publishers to provide direct licences in the same format and the OPLE tools will enable publishers to comply with these requests.

It is also hoped and expected that the availability of OPLE and its use by JISC Collections will encourage library systems vendors to speed up implementation of ONIX-PL.

## Recommendations

1. Version 1 of OPLE is a functioning editing tool, being used by JISC Collections. There are some gaps in the specified functionality and trial use by JISC, publishers and libraries will identify whether or not those extra features are required as well as identify other requirements. Inevitably, a Version 2 will be required and, at that point, further funding will be required for its development. We recommend that , subject to their satisfaction with OPLE, JISC consider contributing towards its further development following this period of trial use.

2. The tender document issued by JISC Collections for an ERM system included the requirement that the system should be capable of supporting at least the following functionality:

(a) Ingest ONIX-PL expressions of JISC publisher licences and store them in a form that will enable the System to extract and process relevant portions of their content as required;
(b) Output ONIX-PL expressions to subscribing institutions without loss of content;
(c) Retrieve ONIX-PL expressions, or requested elements from ONIX-PL expressions, in a convenient human-readable form for JISC staff or the staff of subscribing institutions;
(d) Extract elements from ONIX-PL expressions that are needed to be used for other functions of the System without requiring JISC staff to re-enter or encode them; and
(e) Use the content of ONIX-PL expressions to deliver accurate information on usage permissions at the point of use.

The document also provided a list of usage types that the ERM system would have to reflect. We would recommend that JISC encourages other institutions issuing tenders for ERM systems to adopt similar requirements.

## Appendices:

Appendix 1: Functional review of OPLE Version 1 against the original requirements and design specifications

Appendix 2: OPLE technical development issues