Brockenhurst e-Registers Toolkit (BeRT)

Final Report

Author: *Jon Rowett, Brockenhurst College*
Contact: *Robin Gadd (Project Manager) (rgadd@brock.ac.uk)*
Date: *16/11/2005*
Url: *http://www.brock.ac.uk/projects/bert/*

Acknowledgements

This development project was undertaken by Brockenhurst College and funded by the Frameworks and Tools strand of the JISC E-Learning Programme.

Executive Summary

The project successfully built an open source software toolkit to support the standards-based implementation of a lightweight, web-based attendance registration and monitoring system. The project team used and evaluated a range of tools and a new approach to software development (sometimes called “agile processes”), and this report draws conclusions about their experiences which should be of interest to any college currently pursuing an MLE strategy or engaged in partnerships with other providers.

Background

Brockenhurst College has been using its own in-house developed registration and attendance system (“Emily”) for all Sixth Form classes since September 2003. The system carries details of 2,700 students and typically registers around 1.2 million unique attendances in an academic year. All registers at the college are taken by staff using either desktop PCs in the classroom, or via wireless Tablet PCs. The system also provides individual attendance reports to students and parents via a personalised web portal (“My.Brock”). The system has attracted interest from other Further Education institutions who see the advantages of implementing their own web register systems rather than undertaking large-scale, vendor-led MIS transformation projects which are both disruptive and costly. The project aims to distil this success into an open source software toolkit which would be of real value to Further Education institutions and the wider community as well.

Attendance registration and tracking is of particular importance to Further Education colleges for a variety of reasons, principally meeting funding requirements, payment of Educational Maintenance Allowance to students, and meeting the needs of personal tutors, parents and legal guardians.

Aims and Objectives

- Define a data model (operations, messages and corresponding web service definitions) which enable the production of attendance registers, their completion, storage, editing and retrieval
- Implement the data model in the form of a web service toolkit for the .NET platform: code libraries, data storage, and web service endpoints
- Evaluate a contract-first approach to web service design using current free and commercial tools
- Support e-learning interoperability standards (where appropriate), in particular for the transport of learner, group and timetable data from an existing Management Information System (MIS)
- Develop a client application which can form the basis of a usable registration system

Methodology

The project was managed by the Brockenhurst College MLE Team. Functional requirements were solicited by consulting with a range of staff including lecturers, managers and MIS staff, who commented on the existing Emily system, suggested improvements, and tried out prototypical new user interfaces. Further research was carried out via an open electronic survey which was submitted to several JISC mailing lists and received a good response.

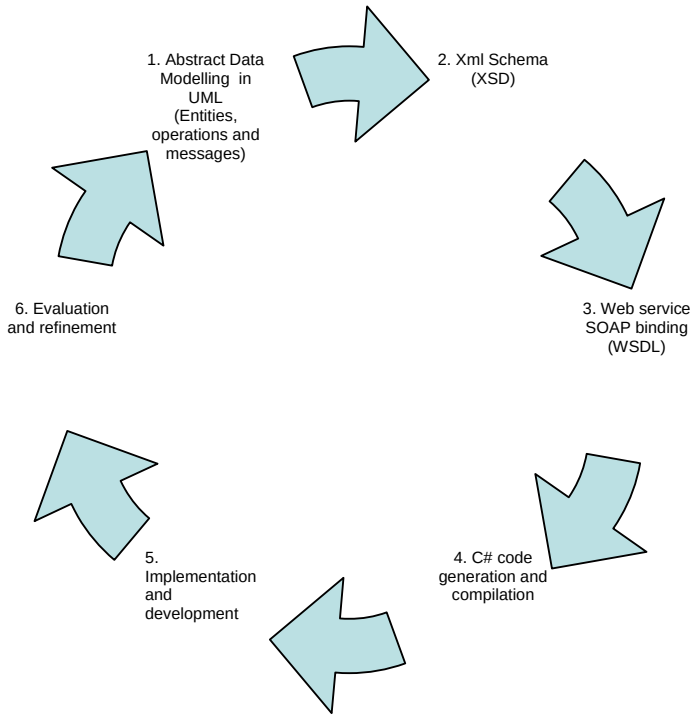
Development Methodology

The team adopted an "agile" approach to implementation whereby small units of functionality were developed, tested and refined using a range of free and commercial tools to manage the software lifecycle. Artefacts were developed in iterations using C# in Microsoft Visual Studio .NET 2003, tested with NUnit[1], built and configured with SourceGear Vault[2] and Nant[3].

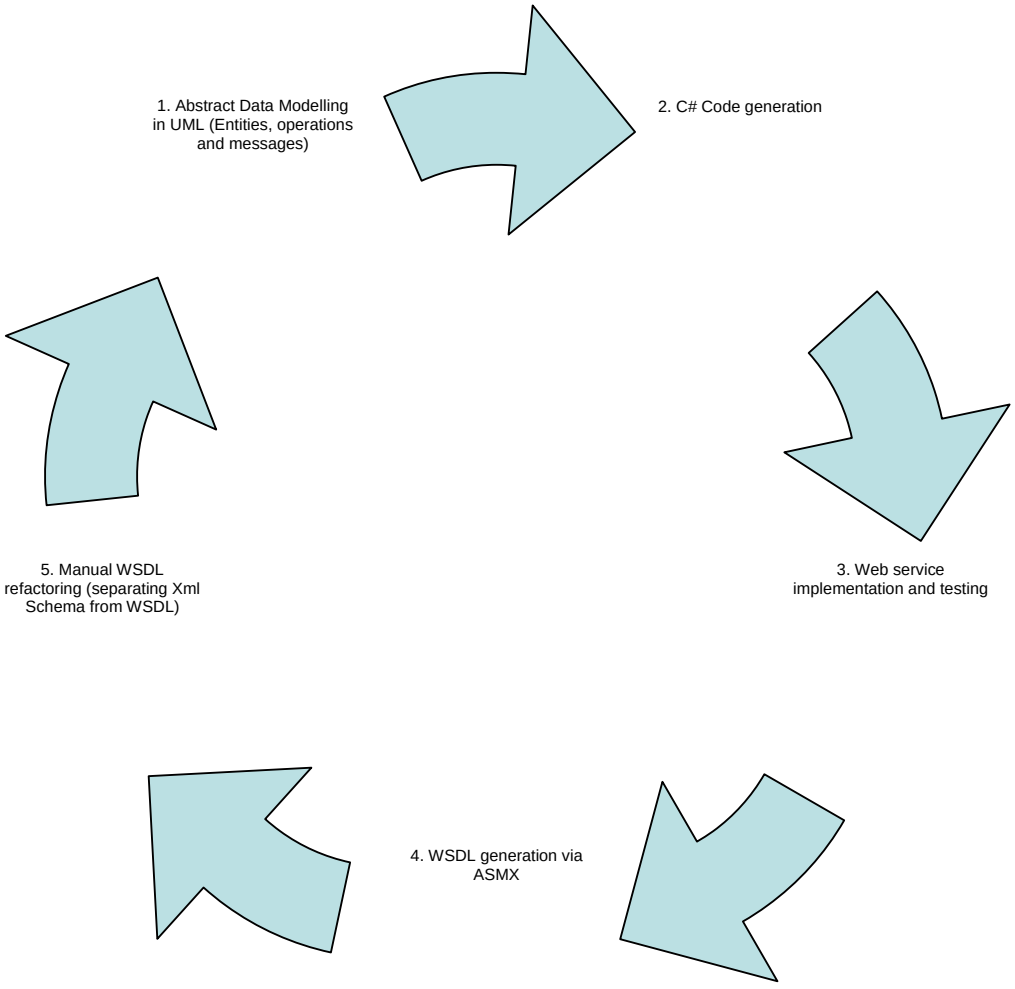
This approach worked well for our small team, enabling us to "just get on with" many aspects of the software development without spending too much time bogged down in process issues or "big up-front design" - while still maintaining an appropriate standard of quality and maintainability. We believe an agile approach to development is ideally suited to development teams (or even individuals) working within Further Education (or similar) institutions, who currently lack any real methodology or process but for whom larger "industrial" processes would be unwieldy, expensive and unnecessary.

The team decided to stick with version 1.1 of the .NET framework, since they did not have much experience with the recently launched version 2.0. In July 2005, two developers attended Microsoft's Tech Ed Europe conference in Amsterdam, and had the opportunity to look in more detail at .NET 2.0, in particular the next generation of web service tools (ASMX and WCF / Indigo), and the next version of the web application framework ASP.NET.

Inspired by a talk given by Colin Smythe on the subject of the IMS[9] process of specification development at a recent Enterprise SIG meeting[10], the team originally hoped to adopt a pure "contract first" approach to iterative web service design:



At the time of the project, some of the web service tools we used, while good, could not be considered mature and were not well integrated. The team evaluated tools such as the excellent WS-Contract First[4], but still felt that this approach carried too much of an overhead for rapid prototyping and development, in particular due to the difficulty of UML-to-Xml Schema-to-C# roundtripping. A compromise approach was taken instead:



It appears that true contract-first design will remain tricky under .NET 2.0 and its related tools, although certainly easier than at present. The continuing development of tools such as WS Contract First[4] and Microsoft's own Distributed System Designer[11] (formerly codenamed "Whitehorse") will continue to evolve and make this style of development more viable for projects such as this.

Implementation

Start-up
The project began with a number of discussion groups with various members of staff. They tried out prototypical user interfaces, and responded very positively to an email-style "inbox/outbox" user

interface which reflected the task-as-message metaphor underlying our favoured approach to service orientation.

An online survey was designed and submitted to the Enterprise and VLE JISCMail mailing lists, the aim of which was to gauge the general level of interest in (and existing use of) electronic attendance tracking systems. (See **appendix A**). Respondents were predominantly from Further Education colleges, although it was not the intent to target the questionnaire at them specifically, which perhaps reflects the degree to which attendance tracking is important to such institutions.

Design
The relevant parts of the existing Emily data model formed the basis for the design of the BeRT model, and core functions of the Emily system (such as generating or submitting a register) were refactored into request/response style messages.

Development
Implementation was straightforward: C# classes were manually mapped to tables and stored procedures in a Sql Server 2000 database – although CodeSmith[5] was used extensively to automatically generate the C# data access layer.

Once the implementation was completed, WSDL was automatically generated (via ASMX) and refactored using XmlSpy[5] so as to separate the Xml schema from the WSDL itself.

It was originally intended that the toolkit would first be deployed "live" at Brockenhurst College as a replacement for Emily, and much work was put into integrating the web service implementation with the College's own MIS. However, due to project slippage (caused by severe IT service disruption over the summer period), the toolkit was not ready in time for the beginning of term and so it was decided that roll-out would be postponed.

In October 2005, close to the end of the project, the lead developer accepted a job offer elsewhere, and so the project was re-scoped to focus solely on the data model and web service implementation parts of the toolkit, at the expense of the example user interface application which was not completed in time.

Documentation
Although the team had not been able to fully realise their desired contract-first approach to web service design and modelling, some degree of satisfaction was derived from generating documentation. Using the Eclipse Workbench[6] and a third party plug-in[7], the Xml schema was transformed into Xml Metadata Interchange (XMI) format, which could then be rendered in Unified Modelling Language (UML) notation using another tool[8]. Recommended further reading: *Automated Generation of Meta Models for Web Service Languages*[13].

Outputs and Results

BeRT toolkit
Available for download at http://www.brock.ac.uk/projects/bert/ includes toolkit implementation source code, Information Model (UML), and web services schema.

Outcomes

Aim:	Outcome:
Define a data model (operations, messages and corresponding web service definitions) which enable the production of attendance registers, their completion, storage, editing and retrieval	The core functionality of a functional electronic registration system was successfully modelled using UML tools and realised as XSD schema and WSDL. The model has been submitted to the E-Frameworks website[12] for comment. We do not expect it to meet the needs of everyone: but we

	hope that our initial work in this area will encourage some discussion and further development within and around the problem space.
Implement the data model in the form of a web service toolkit for the .NET platform: code libraries, data storage, and web service endpoints	A fully functional web service implementation has been produced and released under an open source license. We believe that this can form the solid basis for a variety of different attendance systems, whether web based, smart card, etc.
Evaluate a contract-first approach to web service design using current free and commercial tools	The team evaluated "pure" contract first development before settling on a compromise, and their findings are documented in the "methodology" and "implementation" sections of this report.
Support e-learning interoperability standards (where appropriate), in particular for the transport of learner, group and timetable data from an existing Management Information System (MIS)	The College MIS system was successfully retrofitted to support IMS Enterprise with Timetabling Extensions (using the Sweet.NET toolkit). Custom middleware was written to mediate between BeRT and MIS. This artefact, although perhaps of negligible use to other colleges, has been included in the toolkit for illustrative purposes since it took up a significant portion of developer time!
Develop a client application which can form the basis of a usable registration system	Not achieved in timescale.

Implications

The toolkit clearly needs to be tested in a live pilot in order to evaluate the usability of both the model and the implementation we have provided. The College is currently considering how best to achieve this in time for the next academic year (2006/2007). A possible pilot for the toolkit would be to implement a shared registration service for the local 14-19 Pathfinder Project.

Further development of the BeRT model is clearly needed in order for it to become more widely adopted. There is clearly room in the e-frameworks space for an interoperability specification of this sort, and demand for interoperable attendance systems can only grow as colleges step up their MLE strategies and collaborate more closely with partners and funding bodies.

Recommendations

- A thorough demonstration of the toolkit in a live scenario over a reasonable period of time, to demonstrate the usability of both the model, and the usability and fitness for purpose of our implementation, and to further refine both.

References

1. NUnit, a free unit testing tool: http://www.nunit.org
2. Vault, a commercial software configuration management suite: http://www.sourcegear.com
3. Nant, a free software build tool: http://nant.sourceforge.net

4. WS Contract First: a free web service design tool for .NET and Java:
http://www.thinktecture.com/Resources/Software/WSContractFirst/default.html
5. XmlSpy: a commercial suite of Xml tools: http://www.altova.com
6. Eclipse: an open source Integrated Development Environment (IDE):
7. hyperModel: a free XSD to UML plug-in for Eclipse: http://www.xmlmodeling.com/hyperModel/
8. Enterprise Architect, a commercial UML modelling tool: http://www.sparxsystems.com.au
9. IMS Global, an e-learning interoperability specifications consortium: http://www.imsglobal.org
10. 10th Enterprise SIG Meeting, Chester Zoo:
http://www.cetis.ac.uk/members/enterprise/members/enterprise/meetings/meetingten
11. Distributed System Designer: Graphical application modelling tools included in some versions of Microsoft Visual Studio .NET 2005:
http://msdn.microsoft.com/vstudio/teamsystem/architect/default.aspx
12. The E-Framework (formerly E-Learning Framework): http://www.elframework.org
13. *Automated Generation of Meta Models for Web Service Languages* (Behzad Bordbar and Athanasios Staikopoulos, University of Birmingham) :
http://www.cs.kent.ac.uk/projects/kmf/mdaworkshop/submissions/Bordbar.pdf

Appendix A : E-Registers Survey & Responses

The complete survey resultset is attached below, in CSV format.

DownloadResultsToText.fso.zip