

Sweet.NET Demonstrator Project – ‘Ict4biz’: Final Report

Author: Jon Rowett, Brockenhurst College
Contact: Robin Gadd (Project manager) rgadd@brock.ac.uk
Date: November 16th 2005
Web: <http://www.brock.ac.uk/projects/sweetdemonstrator/>

Acknowledgements

This project was funded by the Frameworks & Tools strand of the JISC E-Learning Programme. Partners were Brockenhurst College (Lead partner), Farnborough College of Technology, and Totton College.

Consulting partners were Capita FHE and Tribal Group.

Executive Summary

The project successfully adapted and used an e-Framework software toolkit (Sweet.NET) to implement inter-college data sharing. 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

The three colleges in the project are partners in a Centre of Vocational Excellence (CoVE), providing ‘ICT for Business’ courses across Hampshire and the New Forest. CoVE managers have, for some time, expressed a need for better access to data relating to provision, achievement and retention, which currently resides in the Management Information Systems (MIS) of the three different colleges.

In the summer of 2004 and with JISC funding, Brockenhurst College developed a software toolkit called Sweet.NET¹ which would allow the exchange of lightweight institutional data in an interoperable and platform independent fashion.

The purpose of this project was to demonstrate the use of this toolkit in a real world scenario by attempting to provide unified access to CoVE data via a web portal, with Sweet.NET handling the back-end integration of the data from the three MIS.

Aims and Objectives

- To implement a partnership deployment of Brockenhurst’s Sweet.NET toolkit
- To “Sweet-enable” the MIS data at each partner college through the authoring of adaptors
- To aggregate data drawn from the 3 MIS into a single partnership portal
- To demonstrate the operational benefits of the prototypical regional MLE
- To test, improve and further develop the toolkit

Methodology

The project was managed by the Brockenhurst College MLE team. Input as to functional requirements was solicited from the Ict4biz steering group, and the MLE team liaised with MIS teams at each college regarding datasets and privacy concerns. The team also consulted with commercial MIS vendors regarding potential licensing requirements, and finding suitable integration points in their systems.

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 NUnit², built and configured with SourceGear Vault³ and Nant⁴.

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.

At the time of the project, some of the tools we used, while very good, could not be considered mature and were not well integrated. In July 2005, two developers attended Microsoft’s Tech Ed Europe conference in Amsterdam, and had the opportunity to explore the next generation of software being made available by Microsoft and their partners. The forthcoming version of Visual Studio, (Visual Studio 2005 Team System), offers tightly integrated “Microsoftified” versions of many of the common tools currently in use (testing, configuration, build automation, versioning, project management), including in-built support for a range of project methodologies both agile and traditional. We believe that these tools will make adopting any software process a more feasible and attractive proposition for those who have yet to do so.

Because we were implementing an existing e-learning specification (IMS Enterprise Services⁵), which we had previously demonstrated to be interoperable¹, we could be confident of interoperability. This “contract first” approach - whereby platform code is generated from a platform agnostic schema – is not well supported by current tools, although perfectly possible, and we note that even the next version of Microsoft’s developer tools still seem to encourage at best a hybrid approach (and at worst, a “code first” approach) to web service development – anathema to interoperability purists - although under .NET 2.0, web services developed in this fashion will be automatically rendered so as to conform to the WS-I base interoperability profile⁶.

Implementation

The project began with discussions between CoVE managers, who all agreed in principle to the project and to commit resources.

The technical team met with representatives of Capita FHE to discuss the implications of integrating data drawn from their Unit-e MIS product into a wider MLE context. The most attractive option, and the one we chose, was to use the in-built Data Warehousing tool to produce frequent exports of institutional data, for import into our own system(s). Direct access to the database was deemed unwise both from a licensing point of view, and also from the point of view of technical complexity. No two college MIS implementations are the same, even if they use the same product: we believe local MIS staff are the people best equipped to extract data from their systems: they are the human integration points and have the best understanding of the data being used.

Significant development work was put into improving the Sweet.NET toolkit. Improvements relevant to the project were:

- Improved error logging
- A better plug-in architecture

The first item is self explanatory. The second improvement means that developers can build “adapter” code libraries (to form the bridge between an existing system and Sweet.NET endpoints) without modifying and recompiling existing Sweet.NET code.

MIS teams at each college were asked to extract data from their systems using their data warehousing software. Here we encountered some issues:

- Significant “time lag” in inter-college communication

- MIS staff not sufficiently briefed about the project by their CoVE representatives
- Data protection concerns at one college meant that all their data was anonymised before being passed to the developers, limiting its usefulness
- One college did not provide any data at all

However, usable data from two different MIS at two colleges was presented, and the MIS staff who produced it are able to now generate these data warehouse exports when required and with minimal fuss. It is anticipated that the third college will replicate the process being used by one of the other partners (whose MIS is essentially the same).

Although we had hoped to automate the process of extracting data and passing it to the servers at Brockenhurst, the communication issues we experienced, combined with difficulties arising from our own network migration work meant that this was not achieved. The reasons for this do not relate to issues of technical feasibility, but rather that this activity is currently a step too far in cultural terms.

The next step was to import the data from the two colleges into a back-end database. The platform chosen was Microsoft Sql Server 2000, and we used its Data Transformation Services (DTS) feature to create routines which will automatically import the data on a repeating schedule. When we do subsequently manage to arrange for data from each college to be transferred on a regular basis, the systems are in place to perform the integration of that data.

In parallel to this effort, database fields were mapped to the IMS Enterprise Services data model, and code libraries were written to enable Sweet.NET to interface with this back-end data store. When the project was originally conceived, we planned to create reusable Sweet.NET adapters for each MIS system. Due to the previously mentioned differences between MIS sites, the adapters we have written are in fact tightly coupled to the individual colleges and the way in which they store information on those systems.

Finally, a web portal was built using ASP.NET which would call the Sweet.NET web services, aggregate and cache the data, and present it to users in the form of various drill-down reports. The portal was made available to CoVE managers and MIS staff on November 15th 2005.

Outputs and Results

New Sweet.NET toolkit

As previously mentioned, a greatly improved version of the toolkit is now available⁷.

Adapters

The adapters are also available from the ict4biz project site (address at top of this report), although we believe they will be of very limited use to third parties beyond illustrative use.

ict4biz Data Portal

The web portal has been made available at <http://emily.brock.ac.uk/ict4biz/portal>. Initial feedback from users has been very positive. When the third college's data is included in the portal the CoVE steering group believe that learner monitoring and reporting process will be greatly enhanced over the current manual data processing.

Outcomes

Objective	Achievement
To implement a partnership deployment of Brockenhurst's Sweet.NET toolkit	Partial success. Technology works, but is only deployed at one site, with participation from two of the three partners.
To Sweet-enable the MIS at each partner college through the authoring of adaptors	Adapting Sweet.NET for use with commercial MIS systems proved to be a relatively easy task. As always, most of the work involves mapping database fields appropriately to the IMS data

	<p>model, and publishing your mappings & vocabulary to all developers of service consumers.</p> <p>We do not believe it is viable to produce generic adapters designed for use at any college (but on a specific commercial MIS) because of the variety of modes of use and customisations made by individual MIS teams.</p>
<p>To aggregate data drawn from the 3 MIS into a single partnership portal</p>	<p>Data returned from the web services is unified and consistent, and is presented via the web portal.</p> <p>Data from one college is anonymous, so may be of limited use beyond general statistical work.</p> <p>Using IMS web services in a scenario where there will only be one service consumer (as with this project, at least so far), adds a significant overhead to the development effort required to produce client applications. However we believe the advantages of web services become more apparent in environments with multiple client applications and development teams, where shielding developers from core databases and encouraging “design for interoperability” are undoubtedly a good thing.</p>
<p>To demonstrate the operational benefits of the prototypical regional MLE</p>	<p>Early feedback from the CoVE steering group indicates that the project will provide considerable efficiency benefits over the largely manual aggregation and data processing methods currently used previously within this partnership and other local/regional initiatives in which colleges find themselves increasingly involved, such as other CoVEs, 14-19 projects etc.</p>
<p>To test, improve and further develop the toolkit</p>	<p>The improved version of the toolkit has subsequently been used (via another set of adapters) for use on several internal projects at Brockenhurst, and is performing well for a range of service consumers used by staff, students and parents.</p> <p>The new adapter functionality works well, with responsibility divided between the adapters themselves (data access & logic), and the core Sweet.NET framework (web service endpoints, logging, error handling, configuration), and we believe that the toolkit has the potential to save developers a great deal of time and effort in implementing IMS Enterprise Services on the .NET platform.</p>

Conclusions

Although our original vision of a multi-site web service deployment had to be somewhat compromised for the reasons described above, overall we consider the project to be a success. We set out to prove the viability of both the toolkit and of inter-college MLE activity, and we have achieved both these aims. We believe that by involving a variety of different users (academic managers and MIS staff) the cultural inhibitions will gradually decline as confidence in the feasibility and security of the web services framework increases.

Recommendations

The continuously changing shape of the FE landscape means that most Colleges can expect to engage in a number of long and short term partnerships with other providers in coming years. Flexible, low cost (and often ad-hoc) integration solutions using well understood technologies, backed up by a responsive MLE development methodology, will be key to the success of these partnerships. We therefore recommend that colleges:

- Look seriously at web services, service orientation and the post client-server world;
- Adopt some form of lightweight development methodology if they have not done so already;
- Work hard to nurture productive partnerships and involve/encourage MIS/IS/IT staff to better collaborate where there are clear benefits for all partners;
- Engage more fully with the e-learning interoperability and e-framework agendas so as to benefit from on-going collective development efforts and ensure that the FE voice is heard.

References

1. Sweet.NET: <http://www.jisc.ac.uk/index.cfm?name=sweetnet>
 2. NUnit, a free unit testing tool: <http://www.nunit.org>
 3. Vault, a commercial software configuration management suite: <http://www.sourcegear.com>
 4. Nant, a free software build tool: <http://nant.sourceforge.net>
 5. IMS Enterprise Services: <http://www.imsglobal.org/es>
 6. WS-Interoperability: <http://www.ws-i.org/>.
- For a complete list of standards supported by .NET web services and WCF (formerly “Indigo”), see <http://blogs.thinktecture.com/cweyer/archive/2005/10/19/414273.aspx>
7. Sweet.NET toolkit download: <http://www.brock.ac.uk/sweet>