Toolkit & Demonstrator Final Report Template (Cut-Down Version)

This template is very loosely based on the JISC Final Report Template available from the JISC Project Management Guidelines web page.

Project name/acronym:	Connecting Smirkboard to Perception via Sweet.net
Project website/blog address:	http://smirkboard.herts.ac.uk/smirkboard_sweet_perception
Report, author(s):	Steve Bennett
Contact person (if different from above)**:**	
Date:	25.10.2006

Methodology

At the beginning we followed a typical waterfall methodology (requirements, system design, unit design etc), but in two ways this proved to be lacking. These were:

- the system design proved much more complicated than we had imagined (the challenge of incorporating external data sources for our system and the consequences this had for its internal data structure) and
- the new interface required to accommodate this and other requirements (for instance the ability to have more than one discussion over the same presentation)

Therefore, approximately half way through the project we switched to a much more *iterative design* methodology. I produced a complete visual basic mockup of the final application which Derek Chassay then went on to implement. This proved to be essential in that mere verbal exchanges about how the interface might look or what functionality should be provided could become bogged down in minutiae – whereas the visual basic prototype gave a much more holistic idea of the eventual destination of the project.

Implementation

Essentially we ploughed most of the early effort into establishing the Sweet.net/Enterprise connectivity. That was a fairly hacky implementation which we considerably improved after the Oxford meeting and I received some particularly useful advice from Carole Shergold from Sussex about how the Enterprise specification defines user level.

At the end, our implementation of IMS Enterprise via Sweet.net was exactly as we had originally planned in the case of student data being passed around. For lecturers however, we had to resort to a less straightforward solution. The reason for this is that Questionmark Perception encrypts lecturer passwords in the database it uses, but does not encrypt student passwords, in practice we found we could directly authenticate students into Smirkboard easily via a web service call to Sweet, which would query the Perception db and give a yay or nay in response. For lecturers, because of the encryption, we had to resort to the artifice of creating Smirkboard users who would become proxies for a Perception lecturer. And that proxy relationship had to be established upon first nominating that particular Sweet.net WSDL endpoint as the source for Smirkboard's user data. Therefore the lecturer would log in to Smirkboard and be authenticated by it, but then would have ownership over those groups and students created by the associated lecturer in Perception.

Outputs and Results

- Zip file of Smirkboard as binary for installation under tomcat
- Zip file of Smirkboard sources
- Zip file of example Sweet.net based Enterprise service + access database to be consumed by Smirkboard

- 3 online presentations about the (a) the project as a whole, (b) a smirkboard tutorial and (c) connecting to Perception/external data source tutorial

Implications

I think the one thing we found that really does need to be addressed in Enterprise specification is the need for a common vocabulary for core groups that are implied in virtually any aggregation of personnel data in a learning and teaching context. Namely there needs to be an agreed name for some mandatory core groups which represent all of any particular category in the InstructorRoleType and the SystemRoleType person specification. Virtually all educational applications will need to have a group composed of ALL the lecturers and another composed of ALL the students. Moreover, most of those applications will involve the former group having some permissions higher than those of the students – after all it is they who will post material for the students to read and digest, post tests, delete comments of a certain nature etc etc. However the developers of such a consuming application need to know when authenticating users, which category to which the person pertains. However, if the name of those groups are left as arbitrary, for instance all the lecturers could be "ALLLECTURERS" "TUTTIPROFESSORI" or "TOUSLESPROFESSEURS" or anything else for that matter – how then could an application consuming that service generically be able to know the role level of a particular user? And if ***generic*** interoperability (that is to say, my ability to write an Enterprise consuming application which requires no implementational tweaks or amendments by the service producer – e.g "make sure your group representing all the lecturers is called 'tuttiprofessori'") is not possible in IMS Enterprise, then why use Enterprise at all?

Conclusions (*optional*)

The benefit of being able to invoke an unlimited number of IMS Enterprise service endpoints as data sources for Smirkboard, means in practice, this one application can effectively create a large number of "domains" where the lecturers see only other lecturers and students within their own domain. While initially we imagined this might be valuable for different institutions to share the one Smirkboard application, now it seems much more likely that a University can offer Enterprise service endpoints for much smaller entities like faculties, departments, degrees or even modules. And it is probably there that the most compelling use of Enterprise technology is likely to be found. For instance if a particular module in a VLE could offer a WSDL end-point representing all the teachers on that module, the groups into which students have been divided for various assignments, then inside Smirkboard, that teacher would be able to log-in and see only his or her own students and not be distracted by seeing a list of names and groups which have no relevance to him/her. This is typically a problem in cohort rather than course based educational applications – (for instance in QM Perception as opposed to in a typical VLE). Smirkboard itself is cohort based, but through this method of using Enterprise, each external data source endpoint becomes in effect a new instantiation of Smirkboard in which the presence of other groups and users is effectively concealed from the inhabiting users. Therefore, with this model in mind, developers can concentrate on creating applications which do particular and specific educational things, without worrying about how it will fit in to a course based system (particularly the case in VLEs).

On the basis of this, and another JISC project with which I was involved, I would make this tentative conclusion. For ***intra***-institutional provisioning of educational applications, IMS Enterprise is a great solution: all you need to do is specify a person and a group wsdl end-points and your application is away! For ***inter***-institutional provisioning however (e.g. for collaboration *between* rather than *within* universities), at least for authentication purposes, I think paradigms like Shibboleth are much better (the trust mechanism means that the one institution is not passing around the other institution's user details). Shibboleth, in my experience however, does require real trust and cooperation (in the human sense) between the teams running the two institutions' service and identity providers, as well as an equality of expertise and system knowhow (in order to do the periodic updates necessary when any other IP or SP joins the web of trust). Enterprise however, beyond the need for agreement of certain core group names, requires no real communication (in a human sense) between the teams running the service provider and those running the service consumer.

Recommendations (*optional*)

Speaking generally (probably at a CETIS level) – I do think there needs to be consideration given to establishing some agreed "dialect" of Enterprise where the names of certain core, potentially even mandatory, groups need to be agreed.