



Project Document Cover Sheet

Before completing this template please note:

- *This template is for completion by JISC funded project managers*
- *Text in italics is explanatory and should be deleted in completed documents.*
- *Please check with your programme manager before completing this form whether they would like to use a specially adapted template specific to your project.*
- *Please see Project Management Guidelines for information about assigning version numbers.*

Project Information			
Project Acronym	Only Connect		
Project Title	Only Connect		
Start Date	1 st August 2008	End Date	31 st July 2009
Lead Institution	University of Bolton		
Project Director	Patrick O'Reilly		
Project Manager & contact details	Chris Frost – 01204 903875, c.frost@bolton.ac.uk		
Partner Institutions	n/a		
Project Web URL	onlyconnectbolton.wordpress.com		
Programme Name (and number)	Institutional Responses to Emergent Technologies		
Programme Manager	Rob Bristow		

Document Name			
Document Title	Final Report		
Reporting Period	n/a		
Author(s) & project role	Chris Frost – Project Manager, Patrick O'Reilly – Project Director		
Date	8 th July 2009	Filename	Finalreport_OnlyConnect.doc
URL	<i>if document is posted on project web site</i>		
Access	<input type="checkbox"/> Project and JISC internal		<input type="checkbox"/> General dissemination

Document History		
Version	Date	Comments
1.0	8 th July 2009	
1.3	1st October 2009	

Project Acronym: Only Connect
Version: V1.2
Contact: Chris Frost
Date: 28 September 2009



JISC Final Report

ONLY CONNECT PROJECT

Author: Chris Frost
Email: C.Frost @Bolton.ac.uk

Date: 28th September 2009

Version: Draft 1.3

Table of Contents

Acknowledgements	3
Executive Summary	4
Background	4
Aims and Objectives	5
Methodology.....	6
Implementation.....	7
Application Architecture	7
Message destination manager	8
Message broker.....	8
Sending email	9
Sending SMS.....	9
RSS	9
Celcat Communications.....	9
Outputs and Results.....	10
The project has produced a number of reusable, interoperable software components and web services. These are:	10
1. The staff communications portal	10
2. Student preference portal	10
3. Central information services	10
4. Message destination manager.....	11
5. Message broker	11
6. RSS Feeds.....	11
Outcomes.....	11
Project experience.....	12
Impact on strategy and development approach.	12
The University has developed a new key e-strategy during the period the project, and this has incorporated some of the lessons from the project. By linking with other University technical projects and providing a potential route for their communication output, the only connect project team has had a greater influence on the use of standard approaches, and in establishing the beginnings of a University wide developer network. This is being incorporated into the E-strategy as a means of achieving a balance between centralized and distributed systems development to meet the needs of our high commissioning layer of users.	12
Impact on the teaching and learning	13
Conclusions.....	13
Implications	13
Building on the project	13
Recommendations	14
References.....	14
Appendix A	15
Appendix B - Communication requirements of the School of Games Computing and creative Technologies	17
Staff communication needs	17

Acknowledgements

The Only Connect project was funded under the JISC Institutional Responses to Emergent Technologies programme.

We gratefully acknowledge the contribution of the following individuals and organisations:

Rob Bristow	JISC
Steve Boneham	Net Skills
Will Allen	Net Skills
Scott Wilson	JISC-Cetis

Executive Summary

To serve the needs of Bolton's diverse student body, we have created a new communications environment with the aim of strengthening the bonds between individual learners and the university through the provision of better support, and learning information. The new learning environment, based on eCommunications and engagement, is part of a wider university change programme, trying to use new technologies and person centred approaches, including personalised learning to address some of the challenges of the changing student body and curriculum.

The new challenges for Bolton's Professional University model are to communicate effectively with students who are not based on campus or who visit only periodically, say for one three hour session per week; who may be based in a work place or studying at a partner institution.

The Only Connect objectives were

1. To create the means of delivering information to students through their preferred channels via a message brokering service e.g. via email, MSN, RSS, or through widget and gadgets
2. To link this messaging service to information systems that produce messages e.g. timetable changes
3. To give students the ability to choose delivery channels for different types of communication

The project took a web services, service oriented approach to creating the system and processes, where components are reusable. The core systems such as the Student records, library or timetable can be linked into the communications environment via connectors. The project sought to establish a connector for the CELCAT timetable to notify students when a room change occurred as an initial exemplar.

This service oriented approach has allowed other services to be developed that were not originally planned for the project. For example a simple messaging portal has been added to allow students or groups of students to be contacted by staff or administrators, without the need to know contact details or preferences. This is also being lined up as an emergency announcement service for critical events – site closure due to swine flu outbreak.

Background

Bolton has a diverse student body with a high percentage of part time, working and work based students. These student have only limited connections to campus, perhaps to attend seminars or visit the library. With the development of the University strategy towards CPD as an important part of the Professional University model, the student head count has increased from 8,500 to 11,500. This increase is in part time students who may only study a single module. These may study with students on full time and part time programmes.

The majority of University processes assume that the student attends campus regularly and so campus based communication is the norm: – notices on notice boards, messages passed on via tutors, notices on doors, University emails and VLE messages. Communication is usually through a single channel, selected by individual academic or admin staff. We recognise that CPD students may not have the same access as other students. There have been occasions when students have driven across Manchester from their work only to discover that a lecture has been cancelled.

The Only Connect project sets out to address the problem by creating a new communications environment to improve students' access to information. Students currently receive a range of communication from the University in the format allowed for by the originating information system e.g.

email, text message, letter. Most of our systems are restricted to the communication channels provided for by the software system. Only Connect provides a communication brokerage that transforms messages issued via one channel to the channel the student chooses for the information type.

The project has developed the communications environment to use web services, enabling plug-ins to be created for different University systems. As a demonstrator of the first application the project has created the means of forwarding timetable change messages from the timetabling system, CELCAT to a message broker service which delivers messages via students preferred channels (sms, email, RSS, widgets). It is anticipated that further developments will incorporate other systems such as the University VLE, library systems and student records system. There are a number of other projects which may well be incorporated into this student portal at a later date. These include a service providing students who have been referred/deferred in a module with make good information.

The project also identified the need for a centralised system for administration and academic staff to send ad-hoc messages to students. This led to the creation of a messaging application that allows staff to message individuals or groups of students without needing to know their contact details. For students, a second portal allows students to select their preferred channels for different types of message.

It became clear during development that the web services developed for this application could also be reused in other University-developed systems. As a result we have started to develop a pool of services with this in mind. We have called this "Central Information Services".

This initial phase of the Only Connect project concentrates on the critical issues of messaging channels. Currently University/Student communication is based entirely on the preference of the communicator with the recipient having to fall into line. So some staff/administrators will send change information through webCT, others through email, or put notices on doors or notice boards. There is no consistency on choice of channels and no one, ideal, uniform way for students to receive communications.

Aims and Objectives

Only Connect aimed to take the communication of personalised timetables and changes as the exemplar to exploit a multi-channel communication strategy where students would choose their personal preference as to how they receive information. The proposed system would be designed and constructed in such a way that it could handle a range of other communications through an open architecture with reusable components

In working towards this goal the aims of the Only Connect project were: -

- To create an Open Messaging system for staff-student communication, accessible on their terms, their technology and at their convenience;
- To provide an online automated information portal for personal and individual timetables utilising user defined technologies, and further enhance the use of the Open messaging system to notify students of changes;
- To develop the components of an Open Messaging system using approaches and open standards that maximise re-use, reconfiguration and sharing
- To investigate the wider potential and limitations of the system in relation to other system generated student communication e.g. the length of content may be suitable for some delivery channels and not for others.
- To identify the system and human implications/effects of the open messaging system as an evaluation of effectiveness of the approach
- To identify systems that may be suitable for future use with the Open Messaging. e.g. E-Vision and the library management system.
- To investigate the sustainability of allowing lecturers to utilize messaging channels such as SMS. Potential issue may include cost of provision and the 'spam' factor.

In the broader institutionally supported eCommunications strategy the university aims to:

- harness the use of personalised technologies and Web 2.0 systems to radically improve student access to learning and academic and personal support;
- model, test and evaluate new generation technology-driven solutions to the issues of student engagement and communication, which informs the university's future drive towards embracing user technologies.

Methodology

The project, used a mixture of traditional requirement gathering techniques to determine the scope and the types of communication that the system was to support. This included investigating the types of communication initiated from the core information systems, and the desired types of communication with students initiated from the school administration and management functions.

The development team interviewed a range of stakeholders including the central administration teams involved in student support and data management; the school admin staff, principle lecturers and student liaison officers, to build use cases of communications with students. An audit was undertaken of the information systems and processes that produce student communications e.g. the student record system prompt to re-enrol, the library system notifications on book loans. A summary of communication types and channels is shown in appendix A. This rich picture was used to inform the design considerations. As an exemplar, the design team concentrated on the process of notifying students of when there was a change in their timetable which seemed at the time to be a straight forward and important communication type.

The system design was based on the principles of service orientation and is made up of a collection of web and database services, including a preference service, messaging services and application connectors. Services are designed to be reusable and to be repurposed. The beauty of this approach is the potential to extend the application the open messaging approach for a wide range of business processes, including incoming communications, by reusing and adding new components. The system developed produces XML messages formatted so that they can be consumed through channels including SMS, RSS and SMTP services. The web applications were written in ASP.NET – mainly in C#. The database of choice is SQL Server 2005.

The student timetable is generated from the University's room booking system, based on Celcat. This system can itself produce messages and dispatch these to students as e-mails. However, the project wanted to investigate whether these e-mails could be captured and transformed e.g. into SMS messages, for example. For this to be tested work was carried out to configure Celcat to be able to issue these messages -this linked in with parallel work taking place to capture student attendance which linked the student to the module event in the Celcat system, through the Celcat attendance module. It transpired that Celcat can e-mail a new PDF file containing the student timetable when a change occurs, a format which rendered it on suitable for transformation. One area of debate concerned the extent to which any messaging service API developed for Celcat should integrate with the messaging system (the use of database triggers etc). A number of strategies were tested including using the email as a simple event trigger.

The University does not have a regular SMS service, although it sometimes uses SMS campaigns in marketing activities, provided by third party companies. As the project progressed more areas, communication processes were identified, which would benefit from the use of channels such as SMS. This included emergency planning in the case of an outbreak of swine flu. To facilitate this kind of communication, the project developed an additional messaging portal.

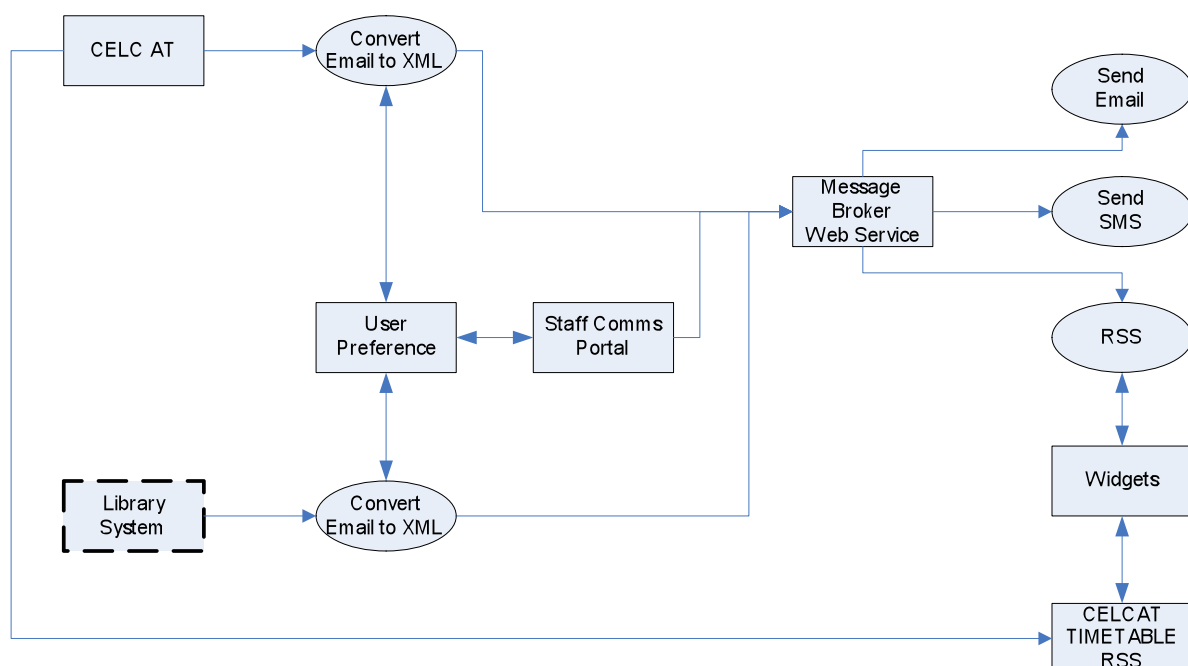
Implementation

The only connect software development was divided into parts and distributed to a small team of developers who worked in a semi agile way. This involved the team developing functional components in cycles using small planning iterations, testing them, learning from the results, and adapting them. The component-based nature of the project lends itself to this type of development and the approach enabled demonstration to stakeholders as the development progressed. This process will also be carried on beyond the end of the project to integrate further communications channels and University systems.

The only connect project segued with a number of other projects around student communication that were going on at same time, based on developments in the school of games, computing and creative technologies. These included projects that are aimed at cohort communication, assessment outcomes and feedback, and make good notification. The only connect project team worked closely with other developers to ensure these would interoperate with the open messaging system. This also had the effect of ensuring that the systems developed in schools, fitted into the architectural principles being applied to central university systems e.g. the use of authentication, security policies and data organisation.

Application Architecture

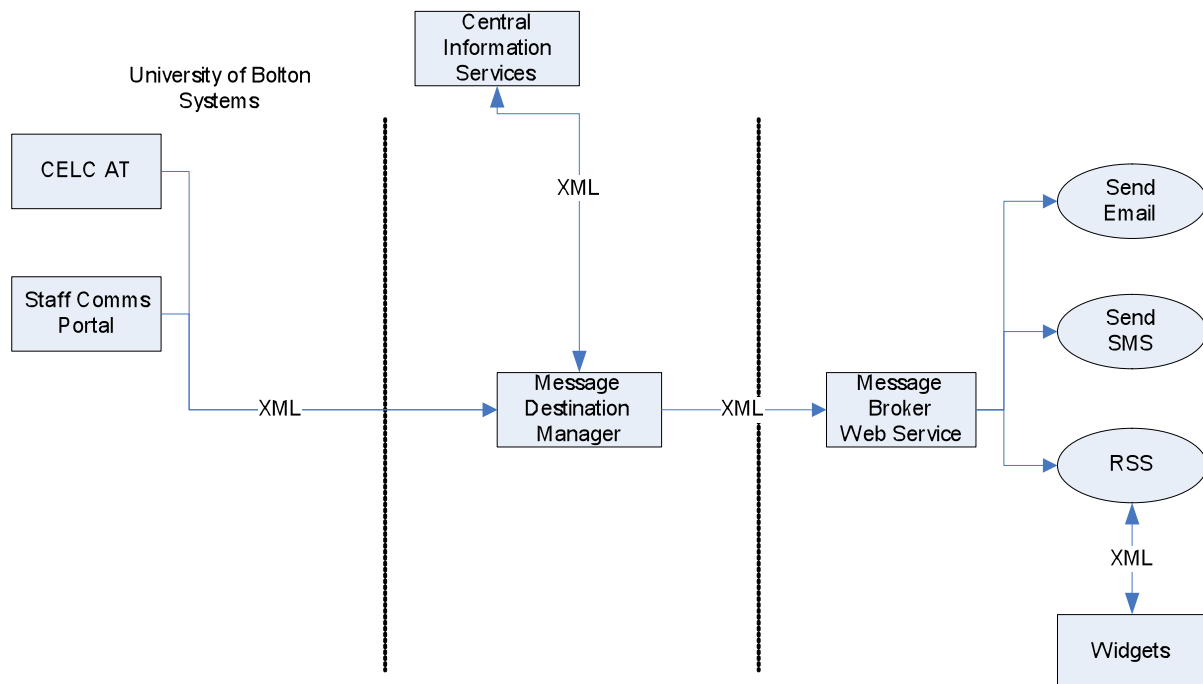
Our initial designs for an application architecture are shown below.



In this initial design the message broker service was the only element of the messaging hub. Each plug-in or home grown application that sent messages would determine the messaging channels that should be used before sending.

Some prototyping took place with the above architecture and through this it became clear that there was a need for a separate function which would determine the destination of messages before

forwarding them to the generic message broker. This service we called the Message Destination Manager and can be seen in the revised design below:



Message destination manager

The message destination manager is a web service that accepts messages from University systems. The only information that it needs includes the various fields that make up the content of the message and the windows user id of the recipient. It is anticipated that in future the system could also be used to communicate with members of staff. It then gathers the required information before forwarding messages to the message broker web service.

The information gathered by the destination manager includes the students email addresses (home and university), their mobile number and their name. The provision of such information is something which we decided would be valuable to many other University web applications. We therefore separated out this function, creating Central Information Services.

During the latter stages of development work on the project news of the threat of swine flu came to the University from government agencies. At this point it was required that the University implement an emergency communications plan. It was decided that a special flag sent to the destination manager would identify messages as emergency messages. In this case messages would be sent out over all available channels.

Message broker

The message broker is designed to be a generic message sender. It requires the following information for each message:

- The message
 - BodyLong
 - BodyShort
 - CostCentre
 - DateTimeCreated
 - Provider (the system sending the message)
 - Subject
 - Type
- The sender

- Login
- MobileNumber
- PersonalEmail
- UniversityEmail

- The recipient
 - Login
 - MobileNumber
 - PersonalEmail
 - UniversityEmail
- The communications channel
 - Id
 - description

It is set up as a web service. Messages sent to this service are stored in a database table. Messages are then picked off at regular intervals by SQL Server jobs that send messages via the prescribed channel. Those which are specified as RSS. Messages sent as SMS or email are flagged once they have been successfully sent.

Sending email

The Sql Server 2005 smtp service, speaks directly to exchange server. Messages that are bound for email are picked up within t-sql. Using this method a history of unsent mail is logged.

Sending SMS

After evaluating a number of SMS service suppliers Meercat Communication was selected because they provided a set of service APIs and billed the service usage retrospectively based on use. Meercat provided an Active X component that we could trigger using a visual basic script and windows scripting host. This script is executed every five minutes using windows scheduler. Each message sent via SMS requires a valid cost centre before it can be sent.

RSS

It was decided that messages should be available to students via RSS and widgets. It was intended that widgets would be developed within the scope of the project but unfortunately this didn't happen. The RSS provision which would be consumed by the widgets is however available. This was developed with .net and allows students access to their own personal feed using a personal key.

RSS feeds provide much richer information than SMS for students with web enabled devices.

Celcat Communications

Celcat, the University's timetabling system was identified as an important system, providing valuable information to students. We decided to use this as an exemplar in this case study.

We were aware that Celcat has it's own built-in messaging service both via SMS and Email. We were keen to harness these outputs for input into our own messaging hub. Initial investigations led to the conclusion that we would have to draw Celcat information into our own database structure in order to trigger messages from timetable changes.

Approach 1

This was deemed necessary as the messaging components built into Celcat have very limited functionality. There are two types of notification available:

- The notifier sends notifications out as soon as timetable changes have been made. The message that is sent is a simple generic message and doesn't give details of the change

that's been made. It simply asks students to look at their new online timetable. This service can be switched off during times of frequent changes such as the start of a new semester.

- The emailer on the other hand is triggered by human interaction. Once the timetabler is happy with the changes he/she has made they click a button to send out notifications. These notifications are more data rich but use Microsoft Outlook on the users machine to send emails out.

Taking a prototyping approach our first move was to create an appropriate database structure into which we set up a scheduled task importing data from the Celcat database. From this database we would trigger data-rich messages sent to our messaging hub from where they would be forwarded to the students preferred channels.

Approach 2

At a later stage in the development of the messaging hub we revisited this area and decided that the logic and functionality of message sending should be encapsulated within the source system and our system should capture its output, forwarding it to our messaging hub. At this point after brief discussions with a developer at Celcat and member of the University networks team we decided to use an SMTP server to capture email notifications from Celcat and transform them into the format required by our message broker.

Outputs and Results

The project has produced a number of reusable, interoperable software components and web services. These are:

1. The staff communications portal

The portal enables academic and support staff to send communications to groupings of students. These may be a module cohort or other groups which the school or service builds. Groups are built from student ID numbers and names – contact details are not required as these are pulled from the student record system SITS. Billing for SMS messages are logged against the cost-code of that department for reconciliation against the SMS providers account.

2. Student preference portal

The student portal allows students to log in and choose their preferred communications channels for different message types. Over time this portal will be extended to provide single sign on access to other student-focussed web applications developed within the University.

3. Central information services

The need for central information services is something which arose as we were developing the application components of this project. It is a central source for web services which are needed across all self-developed University web applications. In the context of the Only Connect project Central Information Services provides information about used by the various components of the application. These include:

- Academic year
- School information
- Subject group information
- Module occurrence details
- Student detail information

4. Message destination manager

The message destination manager receives messages intended for groups of students and checks the user preferences to determine which channels are to be used for delivery. Super users are allowed to send messages to all channels for each student.

5. Message broker

The message broker sends messages to addresses/mobile numbers/rss feeds based on the information sent to it. This is independent from student preferences which are identified in the message destination manager. It was developed in this way to enable use by other applications which do not require the checking of user preferences.

6. RSS Feeds

RSS feeds have been provided which can be accessed by students with a personal key.

It was intended that widgets would be developed within the scope of the project, but time constraints meant that this didn't happen. The next development for only connect will be to build and deploy widgets using Wookie widget server. The University is currently moving to Moodle and incorporating the Wookie widget server. We hope to have widgets ready to plug into the student home page that can render messages and enable students to choose their channel preferences.

Outcomes

Of the objectives we set for the project, the following have been achieved:

To create an Open Messaging system for staff-student communication, accessible on their terms, their technology and at their convenience;	Achieved and capable of delivering messages via a choice of email, SMS and RSS feeds. Widgets consuming data feeds still have to be completed.
To provide an online automated information portal for personal and individual timetables utilising user defined technologies, and further enhance the use of the Open messaging system to notify students of changes;	At the moment students are notified that a change has taken place and are given a URL within the message to connect to their personal timetable.
To develop the components of an Open Messaging system using approaches and open standards that maximise re-use, reconfiguration and sharing	Achieved
To investigate the wider potential and limitations of the system in relation to other system generated student communication e.g. the length of content may be suitable for some delivery channels and not for others.	The Student Preference Portal has to be configured for the message types, limiting the available channels according to the message characteristics. Some communication will need to be restricted to a single channel only - where there is no user choice.
To identify the system and human implications/effects of the open messaging system as an evaluation of effectiveness of the approach	There is already a demand to use the system from the school offices and support units and particular interest in issuing messages to groups. At the moment the system is being rolled out in a controlled way as piloting takes place. There is a danger of the system being over used and loosing effectiveness a real danger of spamming

	students and there may need to be a communication protocol or register.
To identify systems that may be suitable for future use with the Open Messaging. e.g. E-Vision and the library management system.	Achieved
To investigate the sustainability of allowing lecturers to utilize messaging channels such as SMS. Potential issue may include cost of provision and the 'spam' factor.	Not yet achieved. System not yet put into production.
harness the use of personalised technologies and Web 2.0 systems to radically improve student access to learning and academic and personal support;	Partially achieved. Widgets consuming data feeds still need to be completed. RSS is in place. Planned to incorporate services into the Moodle student homepage.
model, test and evaluate new generation technology-driven solutions to the issues of student engagement and communication, which informs the university's future drive towards embracing user technologies.	Partially achieved. Widgets consuming data feeds still to be completed. RSS is in place

Project experience

The only connect project was a technical development project and beyond the requirement gathering and testing, most of the project issues were around technical problems and approaches. With the JISC Cetus service based at Bolton, there has been discussion for some time about the adoption of the e-framework when, like most universities, our core MIS are information, data and process silos. Only connect was an opportunity to both develop a technical solution to solve a communication problem, and an opportunity to test out ideas around service orientation.

Using a service oriented approach was new to the development team, and proved to be a significant learning experience. The temptation was to develop the timetabling solution, based on the .Net SQL databases alone and there was some intense debate about the efficiency and effectiveness of web services and XML for internal systems.

Bolton's core information systems driving messaging do not produce XML messaging, have standard open APIs and associated web services. Therefore, in the case of timetabling, a connector had to be written capable of detecting changes. This proved to be challenging as not all changes should necessarily lead to a message being issued to the student. More recently, both Celcat and tribal have developed web services to improve the interoperability of their systems.

Impact on strategy and development approach.

The University has developed a new key e-strategy during the period the project, and this has incorporated some of the lessons from the project. By linking with other University technical projects and providing a potential route for their communication output, the only connect project team has had a greater influence on the use of standard approaches, and in establishing the beginnings of a University wide developer network. This is being incorporated into the E-strategy as a means of achieving a balance between centralized and distributed systems development to meet the needs of our high commissioning layer of users.

The use of the widget server integrated into the new Moodle VLE gives a potential for a default personal learning environment, with only connect services plugged in. As only connect services can be consumed both inside and outside the VLE, this provides the beginnings of the inside out approach described in the E strategy.

Impact on the teaching and learning

Prior to this project, mechanisms allowing staff-student communication have varied widely across the University with each department using its own methods with various levels of success. The closest any department had to the new system was the computing department old asp page which allowed the sending of email to selected module groups. These communications weren't tracked, but did allow the creation of customised groups of students. Reliability is still dependent on having accurate e-mail addresses stored in the sits student records system, but there was no incentive for this recorded address to be maintained as accurate.

The new system gives students greater control, and a greater incentive to maintain an accurate e-mail address and other contact details. The system also tracks the success of message sending in order to highlight potential problems with data.

At the moment the only connect open messaging system is still being piloted and is due to be rolled out more widely in the new semester. There is a great deal of interest in the system from both the school offices and central support units.

Conclusions

Only connect is the University's first step into service oriented approaches. The University has only a small central development team who have focused on core MIS until now. Most data transfer between systems is carried out in batch. As the University population grows in number and diversity, it will be increasingly important to be able to tailor processes to deal with the variety of student needs. This implies a need to be able to process individual transactions and a need to accommodate individual student preferences. The open messaging system introduces a new development mentality into the University, and the journey to this point has been a challenge for the project team.

Whilst timetabling changes were regarded as an immediate application of particular importance for the majority, part-time students who visit campus only once or twice per week, the technical implementation using Celcat proved more difficult than anticipated, and has taken several iterations of work. However, the supporting open messaging system that has been developed to allow students to choose their channels, the message broker and the destination manager provides a highly flexible messaging infrastructure, that can be expanded, reused and repurposed.

Implications

There are still areas of concern. Whilst the open messaging system consists of a set of web services, it's not clear yet whether there will need to be an underlying infrastructure/architecture to provide the layers of security, version control and performance management expected in a more traditional SOA. This will become apparent as services become more widely used. It may be necessary to look towards an enterprise service bus layer such as the openESB.

Whilst it may seem unsurprising, issues of governance, such as who has access and can use the published services are already causing some concern -easier communication could lead to the abuse and devaluing of communication channels. Other governance and design issues identified were around the importance of designing central web services in such away that they could be reused across multiple applications. It is not hard to imagine a scenario where many services with only subtle differences have been created when one service could have done the job.

Building on the project

It is anticipated that the only connect application will become the messaging hub for other University systems which have communications outputs that talk to students. Other prime systems for connection into the open messaging are:

- Talis, library management system
- The result portal – informing students of make good work that must be completed for referred/deferred modules
- SITS student records system

- The University's new Moodle Virtual Learning Environment

At the moment the only connect student portal consists purely of an interface where students can log in to specify their preferred channels of communication. There are many other applications being developed within the University which have a student focus. Future developments may include integrated these into a one-stop portal for students. This may not be completely realised as E-Vision, the web portal of the SITS student records system, will always exist along side it for students viewing certain information.

An interest in the message broker component has been expressed by the network team who would like to use it to send messages from servers, printers and the like, notifying staff of problems.

Recommendations

As indicated in the conclusion, an area that requires further research is the extent to which a service oriented approach still requires the underlying layers of a service oriented architecture. Those universities, who have adopted an SOA are generally using the likes of IBM's Websphere to provide coordination, performance management, workflow and security. Further research is needed into the open source and open standard Enterprise Service Bus layers such as openESB and glassfish applications server.

References

1. JISC CETIS (2009) Technology Change in High and Further Education – a service oriented approach <http://soa.cetis.ac.uk/soa-pack>
2. Technology Innovation & Development :: Enterprise Service Bus Project <http://tid.ithaka.org/enterprise-service-bus-project>
3. JISC SOA animation (2007) <http://www.jisc.ac.uk/whatwedo/programmes/eframework/soa/>
4. Olivier, W. (2007) Having Your Cake and Eating It: The e-Framework's Service-Oriented Approach to IT in Higher Education | EDUCAUSE CONNECT <http://connect.educause.edu/Library/EDUCAUSE+Review/HavingYourCakeandEatingIt/44596>
5. The open Enterprise Service Bus <https://open-esb.dev.java.net/>

Appendix A

Summary of the type of University to Student Communication

Activity	Message purpose	Current channel
Student recruitment		
Prospectus	Course awareness	Paper brochure
Course database	Course discovery	Website
Course enquiry	Give course details	Leaflet
Open days	Open daily reminders	Postcards, website and posters
Promotional activity	Awareness	SMS campaign
Application receipt	Acknowledgement	Letter, e-mail, UCAS
Application outcome	Confirmation of offer	Letter, UCAS
Enrolment		
Joining instructions	Details of how to enrol, accommodation, etc	Letter, e-mail
School information	Details of initial sessions	Leaflets, website
Notification of fees due	Requests for payment	Letter
Induction		
Course Induction	Course Handbook	Booklet, website
University Induction	Student handbook	Website, diary
Library induction	Resource guides	Leaflets, website
Timetable	Student timetable	Website, e-mail
Online re-enrolment	Reminder to enrolled	E-mail, letter
In course communication (admin)		
Timetable changes	Notification of changes	E-mail, notice boards, notice on the door (no firm system)
Class cancellation	Notification of cancellation	E-mail, notice boards, notice on the door (no firm system)
Module absence	"We missed you"	Telephone, e-mail
Next lesson activity	" Please prepare ...", " please work on assignment..."	Face-to-face
Special event	E.g. guest lecture	Notice board, face-to-face
Module group communication	Exchange of ideas, progress, resources located	VLE, face-to-face, e-mail, web 2.0 services
Assessment	Issue assessments	Face-to-face, Web CT
Assessment reminder	Deadline reminder, Turnitin class ID	Web CT forum, e-mail
Personal tutorial	Appointment time	E-mail, notice board
Assessment feedback	Feedback Marks and comments	Face-to-face
Outcome of exam boards	communicate results	Websites, letter
Remedial action	Assessment make-good sheets	Paper, Web system, letter
Resources		
Library notifications	Book in stock, renewal reminder, interlibrary loan	E-mail

Student support	Ad hoc response to enquiries and problems	Face-to-face, e-mail
Events	Special events of interest to students	Website, posters, flyers
Feedback	Feedback on complaints comments and issues	E-mail, notice board "You said, we did", letter
Password change	Confirmation of change	Telephone, face-to-face, e-mail (no firm system)
Emergency	Notification of closure in the event of an incident	Radio, word-of-mouth, face-to-face, website
Restricted access	Notification of proposed restriction due to event etc	e-mail, website
Graduation	Ticket reminders, invitation	Letter, email
Alumni	Alumni offers, events	email

Appendix B - Communication requirements of the School of Games Computing and creative Technologies

Staff communication needs

Lecturers

Senior and principle lecturers in the school of Games, Computing and Creative Technologies set out the groups with which they would like to communicate.

Communicates with following student groups:

- Individual project students
- Groups of project students – send message to entire group – maybe a web interface that allows lecturers to compile groups of students?
- Module attendees
- Module absentees
- Group of modules including current student from all years who have studied/are studying those modules
- All students in a particular year

Types of communication include:

- Next Tuesday's session will be used to work on assignment – let them know
- Why did absentees not attend?
- Inform of guest lecture applicable to all
- Inform student that they should work on assignments next lesson (SMS deemed more appropriate for this one)

They complained that although the student record system captures the personal email addresses of students there is no user-friendly way of sending to these addresses at present. They are interested in the idea of being able to send via SMS and identified certain communications as being suitable for this method. These include:

- Informing student that they should work on assignments next lesson instead of attending a scheduled class
- Informing students of lecturer absence due to illness
- Prompting students to read detailed email.

Often email communications to the students' University accounts are simply ignored. He observed that students often use messenger products such as MSN and blog sites such as Blogger.

- suggested that staff would value confirmation emails of messages sent out via the different available channels.

School office/student liaison officer

The scenarios provided by the school office were simpler than those highlighted by the lecturers.

Project Acronym: Only Connect
Version: V1.2
Contact: Chris Frost
Date: 28 September 2009

They identified communications with following student groups:

- Module groups
 - Email with cancelled classes
 - Email with timetable changes
- Groups including students on multiple modules
- Occasionally email students grouped by course