Getting Started

Service Oriented Architecture Case Study 1 described how Roehampton University’s in-house Identity Management System had become the key to the provisioning of students with their network account names and passwords, their e-mail accounts and their access to buildings and to borrowing facilities in the Library. The Case study also described, crucially, how the data integration which drove these processes had become very complex and, because of this, the Profiler had become a “risk to the business”.

Therefore, this was a perfect opportunity to do some testing of new ways of developing the computer processes which carry out these key “services” for students following advice in the well thumbed tome “Service Oriented Architecture for Dummies” (1) which declares “Often the first thing a business does in adopting Service Oriented Architecture (SOA) is to take some older applications and transform some of their business functions in to Web Services”.

Moving On

The early stages have now passed. We have discovered that there is, of course, a learning curve associated with using the new development tools, that the development of the computer “code” is no faster and no slower than development in other up-to-date development environments. But testing the operation of a singular “service” is a simpler activity than testing the operation of an element of a larger “module”. Once tested and working, that service will remain working wherever it is deployed as long as the data going into it is correct. What we are left with for this effort is not a singular integration between, for example, the Student Record System (SRS) and the Finance System designed just for that purpose, or a bit of “computer code” which passes data from the SRS to the Timetabling System. What we have a developed, in fact, is a set (a still small set) of re-usable “services”, carrying out distinct, singular business processes, which can be made available for use in a wide variety of different business areas within web based applications. In reality, a set of “Web Services”. These Web Services are: get new student : generate password : generate credentials : generate barcode.

Most people are now familiar with the idea of a “URL” – Uniform Resource Locator, which finds and, within a browser delivers, web pages to a PC. Now imagine that the web page which the URL calls is not an HTML page but a request for data. Your request for that URL could actually produce data from a database and offer it to you for “consumption” within any application you require. It can also be designed, crucially, to offer the data in a standard format which your application would understand implicitly. This format is usually, these days, XML (Extensible Markup Language) which is now an industry standard for data interchange.

Figure 1 shows how these web services are currently structured, “get new student” is a web service which when requested by a URL will perform a data extract from the Student Record System. The data is then passed along the chain of “web services using the URL’s of each service to generate the student login credentials, to generate the student password and to generate the barcode. At this point this new data is “offered “ to the “world”.

Figure 1 – Web Services Deployment First Steps
Realising the Potential
Sudden, the potential of this way of requesting, offering and “consuming” data becomes enormous. Having started in a small way to explore this opportunity in Roehampton, we are now very quickly seeing the potential for large scale gains in simplicity and supportability of existing systems and for the development of additional “services” to students and staff alike.
Quite simply, the opportunities offered by this type of development architecture have not been ignored by the computer industry or, for us in particular, by our major system suppliers.

Figure 2 - Exploring the Potential

Not all these opportunities are available to take just yet, some are ready and capable of deployment, others are being designed, still others are merely thoughts. However the pace at which the key functionality of the ID Management System was redeveloped and implemented holds out great hopes that this really is an efficient and effective way to design, develop and implement new systems.

.............New Problems
So far our internal deployment of Web Services has been simple and straightforward. We are in control of the data which is “offered” from systems like the Student Record System. There are only a few Web Services. We know how the data is defined, we know, for example, our definition of a Registered Student! But it is possible to foresee that there will be a profusion of Web Services and that they will need to be defined, documented and held in a place where they are accessible to future system builders. In other words they will need a “Repository” in just the same way that our business models need to be held in a repository for re-use.
As data is passed much more effectively around our systems then we will need to be sure that the data is always “coherent”, that it is complete and accurate and so we will have to build processes which check the data on the “way out” of databases and on the “way in” to databases.
We know what we use to define a “Registered Student” but do our system suppliers know that also? Will we be able to “trust” the data which comes from such 3rd party Web Services?
We are now beginning to develop a “service” based architecture. That architecture is a close blend of singular computer processes and singular business “services”. We must ensure “that the rules and processes are implemented correctly” (1). Therefore, we must now begin to consider how we will control the growth and exploitation of this new way of doing business. We must begin to explore “SOA governance”.