Dunelm

Services Limited

Consultants in Applied Research & Development for Information & Communications Technology

Session 5: Three Case Studies

Service Oriented Architecture: Preparing Your Business & IT Products

Publication Information

The publication history for this booklet is:

Version	*Date*	*Summary of Amendments*
A	June, 2009	The first release of this booklet.

License

Executive Summary

This fifth session of the Service Oriented Architecture (SOA) series presents three Case Studies. Each Case Study looks at a particular usage of SOA. The first Case Study describes how Oracle have used an interoperability specification based upon Web Services to create an integration portal through which other system can access student information services. The second Case Study looks at how SOA can be used to renovate products by making them Enterprise-capable. This normally requires developing the next generation architecture for the product. The third, and final, Case Study looks at the use of the Amazon Web Services. The available services include application-hosting, e-commerce, cloud computing etc. all of which are accessed using Web Services.

Each of these Case Studies will look at the commercial benefits and outline the technical issues that have to be resolved.

Table of Contents

Publication Information 2
Executive Summary 3
1. Introduction 5
2. Three Case Studies 7
3. Oracle's Enterprise SAIP 9
4. Learning Information Services 13
5. Interoperability using LIS 17
6. Benefits of using LIS 19
7. A Product Ready for Renovation 22
8. The Old Architecture 25
9. The New Architecture 28
10. Benefits of using SOA 31
11. Amazon Web Services 33
12. Using the AWS 37
13. Benefits of using AWS 39
14. In Conclusion 40
Appendix A – Bibliography 44
Appendix B – Acronyms 45
Index 47

1. Introduction

There are many examples of systems based on SOA, in its many forms, already deployed. Also, many suppliers are looking at how their current range of products can either use SOA or be integrated with SOA-based solutions. This isn't to say that these systems make full use of SOA. However, SOA is already being adopted and whether you are a user or supplier, you need an opinion about when and how your organization will adopt SOA.

In Session 5 three case studies are worked through to show how SOA can be used. The learning objectives are to:

- Show how a set of service definitions is used to provide enterprise integration for Oracle's Enterprise Student Administration Integration Pack (SAIP). The SAIP uses an e-learning interoperability specification from the IMS Global Learning Consortium to define how student information, course information, etc. should be exchanged with other related systems using Web Services;
- Discuss some of the different approaches by which SOA was used to drive product innovation in a number of Small/ Medium Enterprises (SMEs). In many cases, making a product SOA-savvy is revolution and not evolution. Typically the new product will have to have its architecture rethought and so this is an ideal opportunity to consider what the 'next generation' of product should also provide. One important consequence of the next generation is that

established clients must be given an easy migration from the old to the new otherwise customers will not make the change preferring to look at alternative solutions;

- Discuss how the Amazon Web Services can be used to extend the capabilities of a system. Amazon provide a number of different commercially available services e.g. cloud computing, e-commerce, on-demand workforce, etc. Many of these are made available through Web Services and so they can be used to realise just a part of a business process as opposed to the whole process.

For each of these Case Studies we look at the commercial benefits and outline the technical issues that have to be resolved. In the Oracle SAIP case study we examine the product architecture, which is a fusion of several products from companies that have been acquired by Oracle during the past five years. The learning information services interoperability specification is described to explain the way in which data is exchanged between the student information service component of the SAIP and other learning systems using Web Services.

In the second case study we present a before and after system architecture for the development of a next generation of product. We describe each of the components in the new architecture and explain how this is essential to support SOA and enterprise integration. In the third case study we describe the Web Services available from Amazon. We explain how these are used and discuss the benefits obtained through their usage.

2. Three Case Studies

Each of the three cases studies has been chosen to demonstrate several of the points raised in this set of materials. The Oracle SAIP case study looks at how a series of products, obtained through the acquisition of companies over a number of years, are integrated. An open, independent third party, interoperability specification is adopted to provide the service interface definitions for key parts of the new product. This set of interfaces is also used to enable third part systems to be integrated with the SAIP. This allows Oracle to encourage third party vendors to provide system extensions thereby increasing the functionality available to the SAIP and making it a more appealing product for deployment in Higher Education.

The product innovation for SMEs case study looks at how an established product is renovated to create the next generation of product. Many deployed industrial products were created between 10 and 20 years ago and have undergone gradual evolution in response to customer demands and technology changes. Many embedded systems have a similar history with the difference, that once deployed, they have an expected life-time of at least 20 years e.g. fire detection and management systems, street lighting systems, etc. For each product there is a time to consider revolution or retirement. The creation of a plethora of wireless technologies, the success of Web–based technologies, the increased sophistication of multimedia content processing and exchange, and the continually decreasing costs

of devices means that a radically different set of new products are now possible. This cases study looks at how these new technologies are used and how enterprise integration is provided using service-oriented computing.

The Amazon Web Services case study looks at how third party services, available as a set of Web Services, can be used. Amazon, who are themselves substantial users of Web Services and grid computing, have created a number of commercial services that either allow a product to be deployed using Web Services or can be used to provide value added services to another service offering. The Amazon approach is appealing because it offers a relatively low cost solution with minimal investment and risk.

3. Oracle's Enterprise SAIP

Over the past five years, Oracle has acquired a number of leading IT companies with key products for the delivery of Web-based enterprise-wide applications. These applications have been used to allow Oracle to re-architecture many of its products based upon SOA. In the higher education sector, Oracle have combined PeopleSoft's (acquired by Oracle in 2005) student information systems with BEA's enterprise service bus (acquired by Oracle in 2008) and their own database technology to create Oracle's Student Administration Integration Pack (SAIP).

The administrative and academic aspects of university life have traditionally been separate. Previous efforts have been made to bring these two aspects of university life together, but there has been no comprehensive solution to bridge the information-flow gap between the administrative back office and the classroom. The SAIP provides this integration. Students not only want access all the time, they also want to participate and collaborate in the teaching and learning environment. They expect the academic and administrative tools they use to deliver an integrated experience and include the latest technologies like portfolios and social networks.

SAIP, released in August 2008, aims to do exactly that with tools that make it possible for higher-education institutions to integrate and manage teaching and learning systems with their student administration systems. A product of Oracle's PeopleSoft Campus Solutions unit, SAIP provides open, standards-based

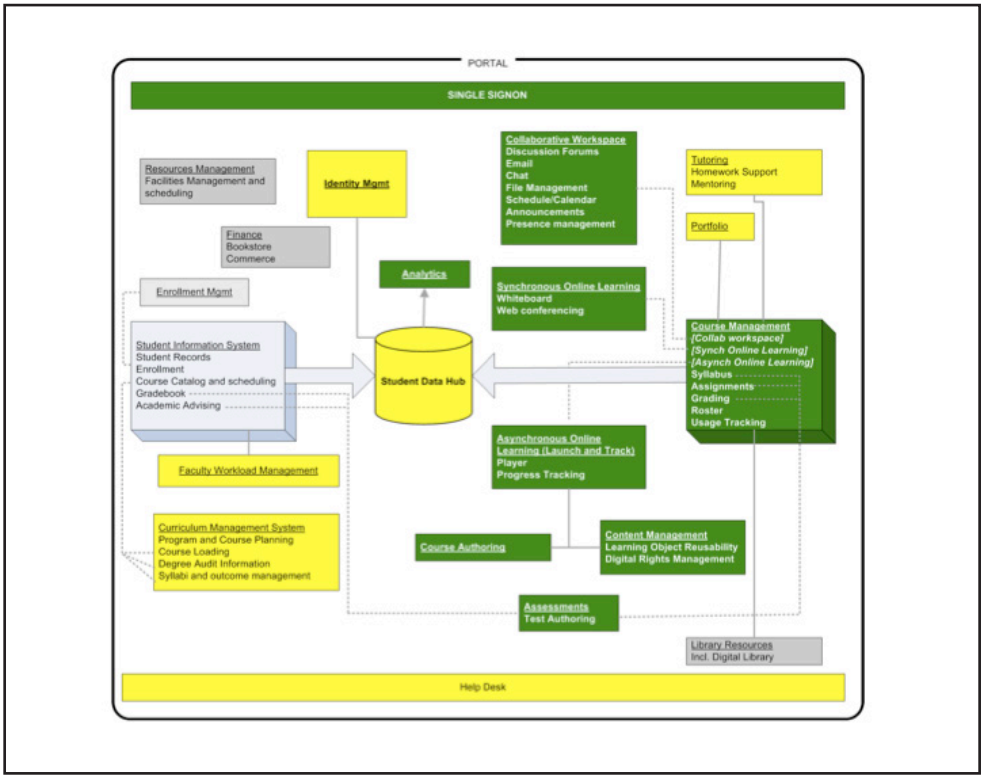

Figure 1 Schematic representation of the Oracle SAIP.

integration with virtually any learning management system, Oracle or otherwise.

Figure 1 is a schematic representation of the Oracle SAIP. The product operates via a Web services-based interface that provides the basis for sharing academic data between the PeopleSoft Enterprise student information system and standards-based learning management systems. When an administrator creates a course in the student system, the interface passes the information to the learning management system, which automatically

creates a corresponding course site or virtual learning space. As students enrol in the course through PeopleSoft Enterprise Campus Solutions, their enrolment information similarly flows directly to the learning management system. The students then receive appropriate access privileges. This design eliminates the need to re-enter data or re-synchronize the two systems with each update.

SAIP is the centrepiece in Oracle's Academic Enterprise Solutions suite, designed specifically for the higher-education market. With the release of additional modules, the suite will integrate and extend all of the systems that constitute the entire education experience. These include student systems, learning management, collaboration and communication; SAIP is the centrepiece.

Perhaps the key to Oracle's strategy was its decision to make the SAIP solution both open and standards based. Doing so provides a platform that institutions can use to advance systems integration efforts, including reining in development costs, choosing best-fit solutions, and providing extensibility for learning management systems. To this end, Oracle partnered with Sakai, Moodle, Desire2Learn and several other vendors to make integration with their learning management systems both flexible and extensible to future technologies, enhancements, and upgrades.

SAIP leverages the IMS Global Learning Consortium's Learning Information Services standard specification, which defines standard information models and protocols for people, courses,

enrolments, and outcomes using Web Services. Even so, in developing SAIP, Oracle recognized that standards limitations still exist. The company is now participating in discussions to update existing standards and create new open standards where none exist.

That reduction in the cost of supporting systems integration projects is achieved in part by providing course information from Oracle's PeopleSoft Enterprise Campus Solutions 9.0. It feeds all of the different applications that need to know what courses exist, along with course details, including learning management systems, portfolios, wikis and facility management systems. The open choice aspect enables institutions to implement solutions that are a best fit to the needs of their constituents. Every application that adheres to the same open standards as SAIP will plug into it — providing the flexibility to change solutions, vendors, processes and to support different processes within the institution. The open and flexible architecture means that upgrades of these solutions can be done independently. Oracle has published the API definition documentation that includes the data model and services definitions needed to integrate custom solutions. With the debut of SAIP and its commitment to open standards, Oracle plans to participate in the community of users and developers and assure maintainability and extensibility of its solutions for higher education.

4. Learning Information Services

The IMS Global Learning Consortium (GLC) creates standards for the development and adoption of technologies that enable high quality, accessible, and affordable learning experiences. IMS GLC is a global, nonprofit, member organisation that strives to enable the growth and impact of learning technology in the education and corporate learning sectors worldwide. IMS GLC members provide leadership in shaping and growing the learning industry through community development of interoperability and adoption practice standards and recognition of the return on investment from learning and educational technology. IMS GLC has approved and published some 20 standards that are the most widely used learning technology standards in the world. Widely used IMS GLC standards include meta-data, content packaging, common cartridge, enterprise services, question & test, sequencing, competencies, access for all, tools interoperability and learning design. These standards have been used widely in higher education, schools/K-12 education and corporate training in regions around the globe.

The Learning Information Services (LIS) specification, from IMS GLC, defines how systems manage the exchange of information that describes people, groups, memberships, courses and outcomes within the context of learning [LIS, 09]. This specification is based upon the aggregation of six component services, as shown in Figure 2. The Learning Information Services specification is implemented using a Web Services

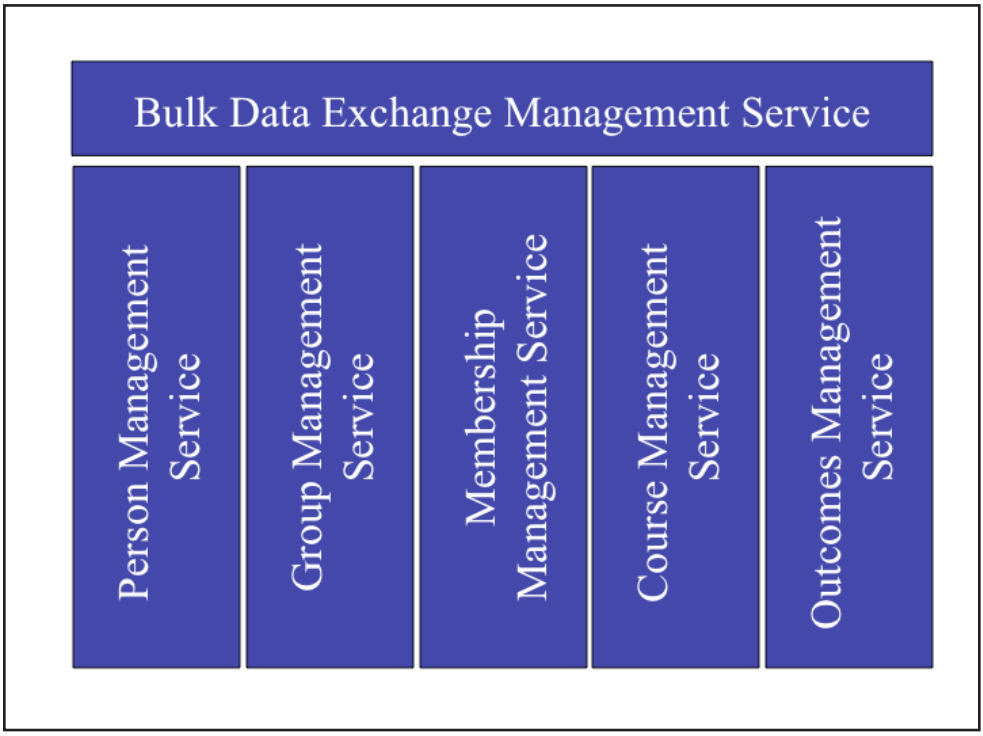

Figure 2 IMS GLC learning information services interoperability.

infrastructure. The specification is based upon the concepts of:

- Interoperability – the LIS focus on the exchange of information between learning information services systems. There are no assumptions in the specification on how the data is managed within the LIS systems;
- Service-oriented – LIS defines the exchange of information in terms of the services being supplied by the collaboration of the systems;

- Component-based – the LIS are composed of the Person Management Service (PMS), Group Management Service (GMS), Membership Management Service (MMS), Course Management Service (CMS), Outcomes Management Service (OMS) and the Bulk Data Exchange Management Service (BDEMS);
- Behaviours and Data Models – the LIS are defined in terms of their behaviours and data models. The behaviours cause changes in the state of the data model and the state of the data model will only be altered as a result of a clearly defined behaviour;
- Multiple Bindings – the LIS information model is to be defined using the Unified Modelling Language (UML) [Fowler, 04]. This enables reliable mapping of the information model into a range of different bindings. The bindings of immediate importance are to the Web Services Description Language and the Lightweight Directory Access Protocol.

The six services act in concert. The Person Management Service allows the manipulation of information about people. The Group Management Service allows the manipulation of groups of people who undertake a common activity and the Membership Management Service defines how people are described as members of a Group or Course. The Course Management Service allows the definition and manipulation of Courses that are available to study and the Outcomes Management Service allows the exchange of grades, scores, etc. achieved as part of

some form of assessment on a Course. The Bulk Data Exchange Management Service is used to initialise two systems and to provide intermediate synchronisation exchanges.

The way in which the set of services are combined is not defined by the specification. Instead profiles of the specification are defined for different domains of usage. The Higher Education profile defines how a Student Information System and Learning Management System use the LIS to support learning in higher education.

5. Interoperability using LIS

A closer look at the SAIP shows that it consists of a 'Student Data Hub' that is used by the Student Information System and the Course Management System. A number of other support systems are also supplied to support analytics, portfolios, synchronous and asynchronous online learning, etc. The LIS specification is used to provide system interoperability by defining what information and how that information is exchanged. The LIS services use Web Services based upon the IMS General Web Services specification and this is a profile of the WS-I Basic Profile (the significance of this WS-I profile is described in Session 6).

For the SAIP, the LIS is used in several ways. In Figure 3 the LIS Web Services provide the interface for communication with the Student Information System (SIS). Firstly, it is used to link the SIS with a Learning Management System (LMS). This enables the LMS to populate its courses with access rights for the various learners from the SIS (using the Person and Membership Management Services). The SIS informs the LMS of the set of programmes, courses and scheduled modules, lectures, etc. using the Course Management Service. The LMS can report outcomes from assessments, tests and quizzes to the SIS (using the Outcomes Management Service). Similarly, the SIS can interact with an ePortfolio Systems using the Person and Outcomes Management Services.

The LIS Web Services can be used by any system i.e. they are

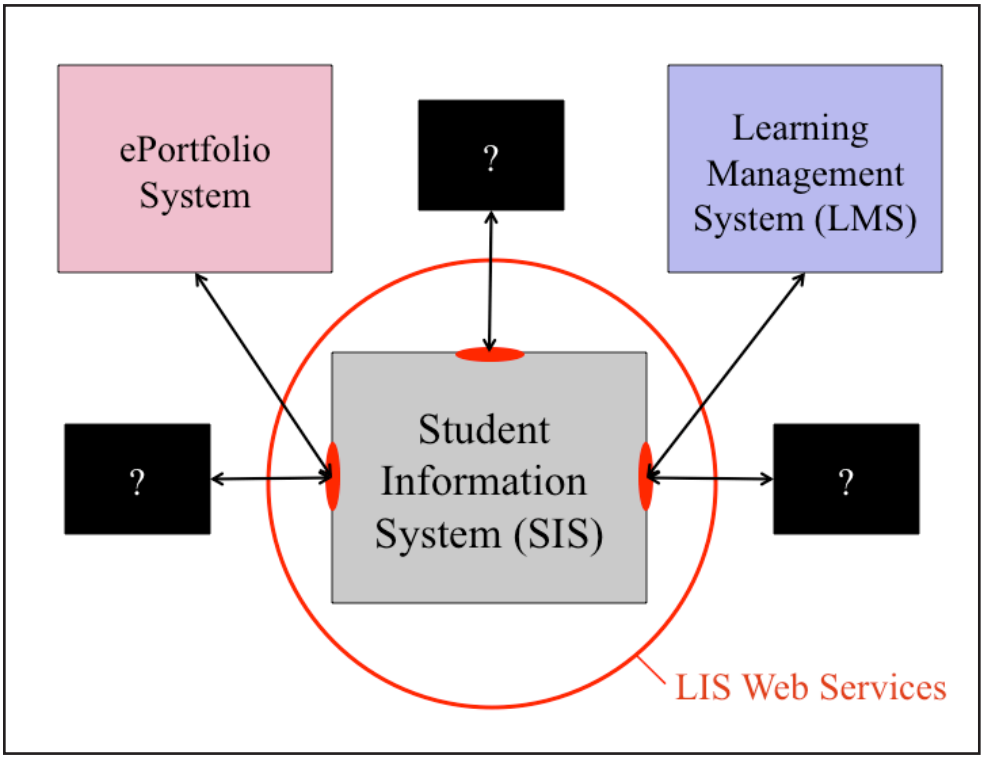

Figure 3 Achieving service interoperability using the LIS.

not limited to SIS/LMS interaction. Therefore, once a system has implemented the LIS specification it can exchange data with any other systems that also support the LIS. However, there is one underlying assumption that must always be considered when integrating such systems. They must have a common set of data semantics i.e. they must have a common vocabulary that defines what is a learner, faculty, parent, course template, a result, etc. If a system has different interpretations then these must never be exposed externally i.e. the system is responsible for mapping between the internal and external realisation.

6. Benefits of using LIS

At the Learning Impact 2008 event (Austin, Texas), senior product managers of the SAIP from Oracle gave a brief presentation on the return on investment when using open interoperability specifications. The starting point was that the use of the LIS avoided having to create a proprietary Web Services interface. The effort used in creating such an interface is substantial, particularly when the work required to advocate and gain external adoption is taken into account.

To give some context, let's consider the effort required to create the IMS GLC LIS interoperability specification. The LIS work took over two years of elapsed time with weekly conference calls, quarterly face-to-face meetings (each lasting two days) and was based on a version 1 Enterprise Services. Over ten different organisations participated in the development from Europe, Asia and the USA. IMS GLC provided the environment for the development and facilitated a lot of the technical work. The specification itself consists of more than ten documents totalling several hundred pages of information model, Web Services binding and best practice recommendations. Clearly, this was a substantial effort.

For Oracle the return on investment justifying the usage of LIS was based upon:

- Reusability – the same Web Services interface was used for several different systems. Each of the Oracle development

partners as well as the in-house development team were able to adopt different profiles of the LIS to support their particular interface requirements;

- Customer satisfaction – a study of customer satisfaction found that they welcomed the usage of open standards. While the details for this were not revealed by Oracle, educated insight suggests that this will be because it avoids supplier 'lock-in' and allows integration/replacement with other third party systems supporting the same open interfaces;
- Reduction of costs for support and maintenance – once the support team understand the LIS specification they have an understanding of the functionality for most of the external interfaces. This makes maintenance easier and it limits the amount of change that is permitted for an interface i.e. its realisation can be altered but the specification of the interface must not be changed;
- Increased sales opportunities – third party suppliers have been able to integrate their products with the SAIP. This has enriched the available functionality from the SAIP thereby increasing the sales opportunities for Oracle and its partners;
- Participation in a broader community – IMS GLC provided an ideal opportunity for discussion on interoperability between the vendors/vendors and vendors/users. This allowed many of the concerns over adoption to be discussed early in the development process and to encourage the user community to commit to including support for LIS in their

invitation to tender documents. Naturally, as a supporter of LIS that gave Oracle and other participants in the development of LIS a commercial advantage.

7. A Product Ready for Renovation

The second case study looks at the issues of renovating a legacy product for Web based usage. This is a typical scenario encountered by any vendors as they seek to re-establish the future of their key products. Many products were originally created in the 80s and 90s as the technology for PCs and network-based approaches were established. A Web-savvy product requires a total rethink for the architecture of most legacy products. The issues that have to be addressed include:

- Technology renovation – products built upon technologies that are only five years old can already have a dated look-and-feel. A new generation must look at the state-of-the-art implementation technology and must address the issues of internationalisation and accessibility, mobility and device independence, web-centric and zero footprint deployment, personalisation and localisation, ease of maintenance and upgrade, and functional extension. Embedded systems should also consider adopting real-time Linux because this will provide a rich environment to improve the functional capability of the embedded system;
- Aging architecture – apart from the implementation renovation, the architecture of the product and product set will also require re-architecting. This is where SOA should be adopted to create a services-based product. Flexibility is

the essential characteristic i.e. the architecture should enable a significant change, over a period of years, in the way the product is delivered, used and its functional capabilities;

- Interoperability – making sure products in the same market sector can be used together without requiring deployment specific implementation. This requires the appropriate use of any sector agreed interoperability standards and specifications for data and services. If these do not exist then all external interfaces should be defined using technology that allows for information exchange between different technology platforms. It may even be necessary to champion, or participate, in the development of new interoperability interfaces. The benefits of such an approach include creating new sales opportunities and enabling integration with products in other market sectors;

- Unification of a set of products – renovation of a product set is an ideal opportunity to look at the underlying implementation technology and to use a common framework for all of the products. In many cases a common platform approach can be used upon which product specific capabilities can be built. If SOA is used then a set of common services can be created e.g. for remote management, report generation, workflow, etc. Software reuse should also be adopted with core functionality built into a set of common libraries cf. XML document handling libraries etc.

- Customer integration requirements – the integration aspirations of the customers must be addressed. These will

also inform the technology and architectural renovation requirements. For enterprise systems integration it is important to identify the order of priority for third party products;

- Use of third party services – the functionality of a product can be extended by integrating it with third party services or by using third party delivery capabilities e.g. using the Amazon cloud computing infrastructure (see the next case study). This can provide a quick and cost effective way to develop and deploy a new product. At a later date the third party solution can be replaced by a better or cheaper alternative from either another third party or developed in-house.

8. The Old Architecture

There are many examples of legacy products requiring renovation. The example we discuss at in detail is that of an embedded system that is used to control an array of remote sensors linked to the controller using a set of sensor loop networks. The set of sensors supply a range of information on the ambient environment. This example is taken from a real piece of innovation work that was started in late 2008. The control panel was designed some 20 years ago, contained a traditional embedded real-time operating system and consisted of an array of physical buttons and switches with a simple monochrome character display. A local management system was connected, via RS232, to each control panel (a schematic representation is shown in Figure 4). There were limited sales in other countries but each country required a specially manufactured version of the control panel.

Most business premises have one of these, or similar, control systems. Each product must undergo a daunting set of certification tests to show that the system is highly reliable and conforms to all of the relevant international standards. The lifetime of such a system is between 20-30 years minimum. The limitations of such a system include:

- There is no remote management or maintenance capability. Upgrades and testing of the system require a physical visit;
- There is no easy way to manage controllers located in

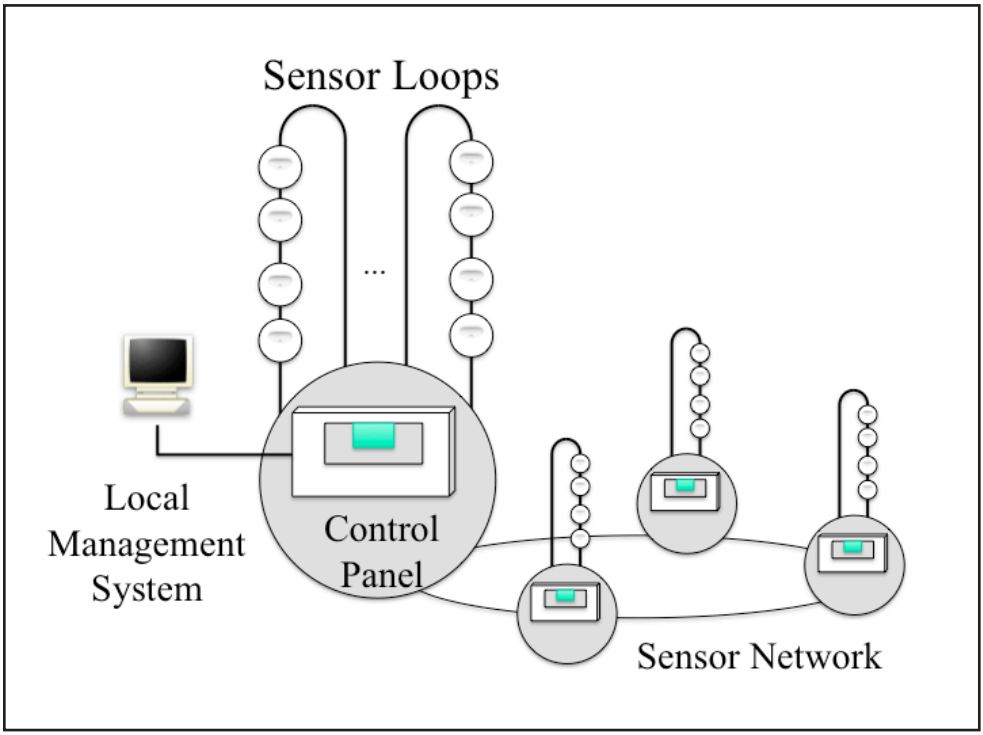

Figure 4 The original system architecture.

different buildings. Also, it is not possible to interconnect these controllers to other building/facility management systems;

- There is no software-based internationalisation capability, the character-based display provides limited status/control information and the host operating system/processor cannot drive many of the more recent networking/storage devices;
- The host processor provides limited data processing and cannot support sensors that send image/audio information.

While the product sales were still healthy, there was increasing competition from systems that were more enterprise-oriented. The innovation brief was to re-architect the product range (the control panel and the sensors) to increase national and international market share and to remain competitive over the next decade at least.

9. The New Architecture

The innovation is focused on two components in the control system: creation of the new remote management system and the new controller itself (the new architecture is shown in Figure 5). The controller was redesigned such that:

- A new host processor was selected, configured to support real-time Linux and supported using a substantial external storage. This made it possible to host a full Web server plus the new set of control applications. All of the control software was re-implemented and turned into a set of services;
- Access to the Internet through wired and wireless network access was included using an integrated remote access router;
- A new user interface was developed based upon Web browser access (to give the same experience for local/remote access through any form of browser) and using a multi-colour touchscreen display. The interface was implemented using cascading style sheets thereby simplifying internationalisation and accessibility requirements;
- A set of Web Services interfaces were defined and implemented to support the new services, provide communication with the remote management system, provide enterprise workflow integration, provide enterprise facilities management integration and to enable remote real-time interaction including the multimedia, voice and video data

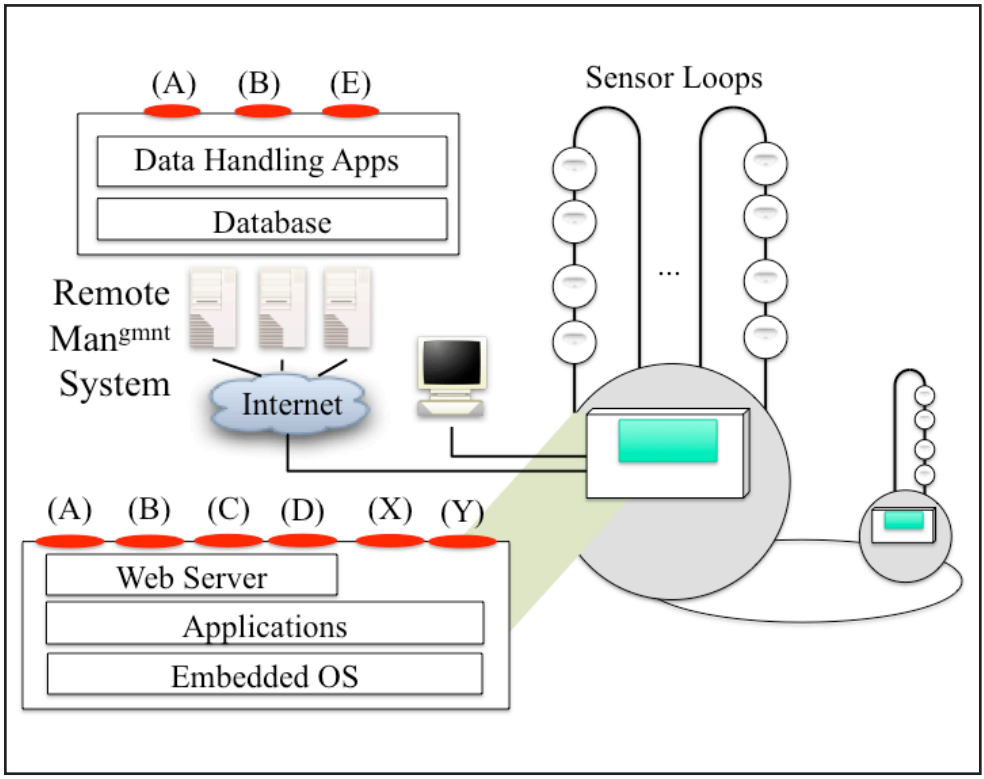

Figure 5 New system architecture using a services approach

exchange (denoted as 'A', 'B', 'C', 'D', 'X' and 'Y' in Figure 5).

The new remote management system was designed to improve maintenance and to support enterprise integration. Multiple control systems can be managed from a single point of management. This management centre also acts as a data collection, collation and analysis point. Data is collected from the controllers and their set of sensors. This data is analysed

to determine the operational capability of the sensors and to provide preventative maintenance. The appropriate operational management reports are created and made available through the enterprise integration interface. The remote management system is also used as the distribution point for any software/firmware upgrades.

The Web and Internet centric approach allows the system to be turned into a set of co-ordinated services. The use of real-time Linux enables standard Web server tools to be employed thereby reducing development time, while improving reliability and flexibility. New functionality can be supplied as part of an upgrade or downloaded as new services. The browser-based display significantly reduces the cost of manufacture for different parts of the World, particularly for Asia and the Middle East.

This provides an excellent basis for this new generation of product.

10. Benefits of using SOA

The benefits of adopting SOA as part of this renovation are:

- All of the external interfaces, realised using Web Services, provide a clear service interface definition. Any decision to make these open interfaces i.e. available for third party integration, is reduced to a commercial justification. The implementation technology provides platform interoperability for the services and so no further development work is required. This means that third party integration can either be promoted through an open interface or through one-to-one commercial integration agreements. In fact, both approaches can be adopted for different interfaces. Again, the approach should be determined by the business strategy;
- Support for enterprise and workflow integration means that the product's services can become part of a larger business process or the product itself can drive more sophisticated business processes. This allows a client to create complex processes that fits their specific business process requirements. In turn, this places the product at the heart of those new business processes thereby making it an increasing important product to retain and requiring a maintenance contract;
- The services approach allows the remote management component to be deployed in a variety of ways e.g. managed services, software as a service, etc. The use of Web-based

technologies simplifies user access to the control system (through a browser) and so enables a range of remote access capabilities.

The key benefit of this service-oriented approach is that the functionality of the product can be easily extended, deployed and maintained. This will significantly extend the lifetime of the product set and provide a range of new market opportunities.

11. Amazon Web Services

Since 2006, Amazon Web Services (AWS) has provided companies with access to an infrastructure Web Services platform in the cloud. AWS provides access to compute power, storage and other services providing a suite of on-demand elastic IT infrastructure services. The AWS supports a wide range of development platforms or programming models. There are no up-front expenses or long-term commitments with payment only for what is used. This makes AWS a cost-effective way to deliver applications. Through AWS, an organisation can take advantage of Amazon.com's global computing infrastructure, that is the backbone of their $15 billion retail business and transactional enterprise whose scalable, reliable, and secure distributed computing infrastructure has been developed and refined for over 13 years.

Amazon Web Services delivers a number of benefits for IT organizations and developers alike, including:

- Cost-effective – pay only for what is used and as used with no up-front commitments. As the Amazon Web Services cloud grows, the operations, management and hardware costs shrink, and Amazon claim to pass the savings onto the on-demand users;
- Dependable – the AWS web-scale infrastructure is well tried and tested in even the most extreme loading conditions. The AWS cloud is distributed, secure and resilient, providing reliability and massive scale;

- Flexible – any platform or any programming model can be used for an application. The AWS resources used by an application can be controlled to fit the target costs;
- Comprehensive – AWS provides a number of services that can be incorporated into an application. These services, from databases to payments, can support the creation of a wide range of applications cost effectively and with less up-front investment.

The available set of Amazon Web Services are categorised as:

- Infrastructure services – the set of on-demand resources including access to the Elastic Compute cloud (Amazon EC2), SimpleDb, Simple Storage (Amazon S3), CloudFront, Simple Queue (Amazon SQS) and elastic MapReduce;
- Payments and billing – services that provide a managed billing and payments infrastructure. The available services are the Flexible Payments Service (Amazon FPS) and DevPay. This removes the need to establish our own ecommerce infrastructure;
- On-demand workforce – this is the Mechanical Turk service that enables an organisation to access thousands of high quality, low cost, global and on-demand workers. Mechanical Turk enables results of their work to be integrated directly into the appropriate business processes and systems;
- Alexa Web Services – this provides access to the Alexa Web information services. The Alexa services provide information

on Web traffic metrics, top site lists, site demographics, etc. Amazon provide access through a set of Web Services;

- Merchant Services – currently this consists of the Fulfillment Web Service (Amazon FWS). It allows merchants to access Amazon.com's fulfillment capabilities through a simple Web Services interface. Merchants can programmatically send order information to Amazon with instructions to physically fulfil customer orders on their behalf.

Typical uses of AWS are:

- Application, Web and media hosting – to support hosted applications and provide software as a service;
- Backup and storage – external remote backup and storage;
- Content delivery – remote storage and streaming of on-demand content;
- E-commerce – payment and billing capability;
- High performance computing – external on-demand compute capability;
- On-demand workforce – link together and electronically integrate an external work workforce;
- Search engines – support for hosted search engine and Web metrics provision and analysis.

Apart from Amazon there are other suppliers of similar services e.g. eBay, Google etc. They all provide access to a high

performance compute infrastructure through Web Services with a number of tailored business services. They are an excellent way to deploy, quickly and cost effectively, a new business service.

12. Using the AWS

Using the Amazon Web Services is very simple. First visit the web-site http://aws.amazon.com and sign-up for a free AWS account. You can use a previously created Amazon account but some extra information is requested. Once the account has been created you are assigned an 'Access Key ID". This access key is used to uniquely identify you and must be supplied to the requests sent to the Amazon Web Services. It is recommended that a second access key is created to rotate their usage with the Web Services. Each access key has an associated 'Secret Access Key' and together these are used to generate signatures for Web Services authentication.

The next step is to review the extensive set of technical documentation and support tools. Each service has its own set of documentation and recommended tools. For example, the Elastic Compute Cloud has eight tools ranging from the Amazon EC2 AMI Tools (command line utilities to help bundle an Amazon Machine Image and to upload the image) to the AWS Toolkit for Eclipse (a plug-in for the Eclipse Java IDE that makes it easier for developers to develop, deploy and debug Java applications using Amazon Web Services). Further available information includes sample code, tutorials, public data sets and access to user groups.

For each service there is an associated service level agreement and pricing structure. These need to be carefully reviewed and assessed, particularly the pricing structures. The pricing

structures cover data loads in and out of the service as well as CPU usage, etc. of the service itself. So the actual costs for a service will be composed of several pricing strands. Amazon provide a 'Simple Monthly Calculator' which should be used to establish a rough estimate of the price (this tool ensures that you do not accidently leave out any of the pricing strands).

The final stage of preparation before starting the implementation is to review the set of APIs and WSDL files (more about both of these in Session 6). These files simplify integration with the appropriate AWS and make it easy to start using the relevant service. Experience has shown that using AWS, or the equivalent offerings from Google, etc. will significantly reduce the time for launching a new service. For an SME such an approach is particularly attractive and effective way to launch new services.

13. Benefits of using AWS

There are two clear benefits of using the AWS or equivalent services:

- First and foremost the use of third party infrastructure Web Services provides a considerable saving in the cost of ownership. Amazon claim that there is a factor of 10 saving when using their services as opposed to establishing an equivalent infrastructure. A further benefit is that the on-demand approach allows the services and corresponding costs to be scaled as the usage requirements increase. This reduces the cost of set-up and ensures that the quality of service can always be tuned to the supply requirement;
- Secondly, it allows the development team to focus on the unique features of the new web-based applications as opposed to expending effort on standard infrastructure features. Furthermore, new flavours of an application can be quickly created through integration with different infrastructure Web Services. This increases the range of business services that can be made available while reducing time to deployment.

As the range of requirements on the infrastructure increase it is always possible to move to a dedicated infrastructure. However, usage of third party Web Services will, in general, be the cheapest and quickest approach for the development and deployment of new business services and applications.

14. In Conclusion

There are many examples of systems based on SOA, in its many forms, already deployed. Also, many suppliers are looking at how their current range of products can either use SOA or be integrated with SOA-based solutions. This isn't to say that these systems make full use of SOA. However, SOA is already being adopted and whether you are a user or supplier, you need an opinion about when and how your organization will adopt SOA.

For each of the Case Studies we focused on the commercial benefits and outlined the technical issues that have to be resolved. In the Oracle Student Administration Integration Portal (SAIP) case study we examined the product architecture, which is a fusion of several products from companies that have been acquired by Oracle during the past five years. The learning information services interoperability specification was described to explain the way in which data is exchanged between the student information service component of the SAIP and other learning systems using Web Services. In the second case study we presented a before and after system architecture for the development of a next generation of product. We described each of the components in the new architecture and explained how this was essential to support SOA and enterprise integration. In the third case study we described the set of Web Services available from Amazon. We explained how these are used and discussed the benefits obtained through their usage.

For Oracle, Case Study one, the return on investment justifying

the usage of an open interoperability specification (the IMS GLC Learning Information Services) was based upon:

- Reusability – the same Web Services interface was used for several different systems. Each of the Oracle development partners as well as the in-house development team were able to adopt different profiles of the LIS to support their particular interface requirements;
- Customer satisfaction – a study of customer satisfaction found that they welcomed the usage of open standards;
- Reduction of costs for support and maintenance – once the support team understood the LIS specification they had an understanding of the functionality for most of the external interfaces. This made maintenance easier and limited the amount of change permitted to an interface i.e. its realisation could be altered but the specification of the interface could not be changed;
- Increased sales opportunities – third party suppliers have been able to integrate their products with the SAIP. This has enriched the available functionality from the SAIP thereby increasing the sales opportunities for Oracle and its partners.

The benefits of adopting SOA as part of product renovation, Case Study two, were:

- All of the external interfaces, realised as Web Services, provided a clear service interface definition. Any decision to make these open interfaces i.e. available for third party

integration, was reduced to a commercial justification. The implementation technology provides platform interoperability for the services and so no further development work is required. This means that third party integration can either be promoted through an open interface or through one-to-one commercial integration agreements;

- Support for enterprise and workflow integration means that the product's services can become part of a larger business process or the product itself can drive more sophisticated business processes. This allows a client to create complex processes that fits their specific business process requirements. In turn, this places the product at the heart of those new business processes thereby making it an increasing important product to retain and requiring of a maintenance contract;

- The services approach allows the remote management component to be deployed in a variety of ways e.g. managed services, software as a service, etc. The use of Web-based technologies simplifies user access to the control system (through a browser) and so enables a range of remote access capabilities.

The benefits of using the Amazon Web Services, Case Study three, or equivalent services were:

- First and foremost the use of third party infrastructure Web Services provides a considerable saving in the cost of ownership. Amazon claim that there is a factor of 10

saving when using their services as opposed to establishing an equivalent infrastructure. A further benefit is that the on-demand approach allows the services and corresponding costs to be scaled as the usage requirements increase. This reduces the cost of set-up and ensures that the quality of service can always be tuned to the supply requirement;

- Secondly, it allows the development team to focus on the unique features of the new web-based applications as opposed to expending effort on standard infrastructure features. Furthermore new flavours of an application can be quickly created through integration with different infrastructure Web Services. This increases the range of business services that can be made available while reducing time to deployment

Appendix A – Bibliography

[Fowler, 04] *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, M.Fowler, Addison-Wesley, 2004, ISBN: 0-321-19368-7.

[LIS, 09] *IMS GLC Learning Information Services Specification V1.0*, L.Feng, B.Lee and C.Smythe, IMS Global Learning Consortium, July 2009. http://www.imsglobal.org/specifications.html

Appendix B – Acronyms

AMI	Amazon Machine Image
API	Application Programming Interface
AWS	Amazon Web Services
BDEMS	Bulk Data Exchange Management Service
CMS	Course Management Service
CPU	Central Processing Unit
GMS	Group Management Service
IDE	Integrated Development Environment
IMS GLC	IMS Global Learning Consortium
IT	Information Technology
LIS	Learning Information Services
LMS	Learning Management System
OMS	Outcomes Management Service
PC	Personal Computer
PMS	Person Management Service
ROI	Return On Investment
SAIP	Student Administration Integration Portal

SIS	Student Information System
SLA	Service Level Agreement
SME	Small/Medium Enterprise
SOA	Service Oriented Architecture
UML	Unified Modelling Language
WSDL	Web Services Description Language
WS-I	Web Services Interoperability Organisation
XML	Extensible Mark-up Language
XSD	XML Schema Definition

INDEX

A

Amazon Machine Image. *See* AMI

Amazon Web Services. *See* AWS

AMI 37, 45

API 12, 45

Application Programming Interface. *See* API

AWS 4, 33, 34, 35, 37, 38, 39, 45

B

BDEMS 15, 45

Bulk Data Exchange Management Service. *See* BDEMS

C

Central Processing Unit. *See* CPU

CMS 15, 45

Course Management Service. *See* CMS

CPU 38, 45

E

Extensible Mark-up Language. *See* XML

G

GMS 15, 45

Group Management Service. *See* GMS

I

IDE 37, 45

IMS GLC 13, 14, 19, 20, 41, 44, 45

IMS Global Learning Consortium. *See* IMS GLC

Information Technology. *See* IT

Integrated Development Environment. *See* IDE

IT 1, 9, 33, 45

L

Learning Information Services. *See* LIS

Learning Management System. *See* LMS

LIS 4, 13, 14, 15, 16, 17, 18, 19, 20, 21, 41, 44, 45

LMS 17, 18, 45

O

OMS 15, 45

Oracle's Enterprise SAIP 4, 9

Outcomes Management Service. *See* OMS

P

PC 45

Personal Computer. *See* PC

Person Management Service. *See* PMS

PMS 15, 45

R

Return On Investment. *See* ROI

ROI 45

S

SAIP 4, 5, 6, 7, 9, 10, 11, 12, 17, 19, 20, 40, 41, 45

Service Level Agreement. *See* SLE

Service Oriented Architecture. *See* SOA

SIS 17, 18, 46

SLA 46

Small/Medium Enterprise. *See* SME

SME 38, 46

SOA 3, 4, 5, 6, 9, 22, 23, 31, 40, 41, 46

Student Administration Integration Portal. *See* SAIP

Student Information System. *See* SIS

U

UML 15, 44, 46

Unified Modelling Language. *See* UML

W

Web Services Description Language. *See* WSDL

Web Services Interoperability Organisation. *See* WS-I

WSDL 38, 46

WS-I 17, 46

X

XML 23, 46

XML Schema Definition. *See* XSD

XSD 46

This work was funded by Digital 20/20. Further information is available at http://www.digital2020.org.uk.

This booklet was written and produced by Colin and Christine Smythe of Dunelm Services Ltd (http:///www.dunelm.com), at 34 Acorn Hill, Stannington, Sheffield, S6 6AW, Tel: 0114-2334009, E-mail: colin@dunelm.com.