All Part of the Service? Addressing Data Longevity in Service Oriented Architectures

Jenny Leonard

Business Information Systems, Faculty of Economics and Business, University of Sydney, j.leonard@econ.usyd.edu.au

Follow this and additional works at: http://aisel.aisnet.org/acis2009

Recommended Citation
http://aisel.aisnet.org/acis2009/7

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
All Part of the Service? Addressing Data Longevity in Service Oriented Architectures

Jenny Leonard  
Business Information Systems  
Faculty of Economics and Business  
University of Sydney  
Sydney, Australia  
Email: j.leonard@econ.usyd.edu.au

Abstract

Software as a Service is gaining increasing pre-eminence as a means of delivery of software for business processes. Indeed, as standards develop in this area, the possibility arises that web based Service Oriented Architectures will develop allowing businesses to source services from multiple providers, and integrate them as required. The positive implications of these developments for supporting business flexibility have been widely discussed. However, there may be other business needs that are less well served. In particular, the need to support long term, sometimes permanent data – here termed data longevity- may not be well addressed. This paper reviews literature to identify seven requirements for data longevity, and to assess how well service oriented architectures fulfil those requirements. It finds gaps, and suggests areas for future research.

Keywords

Software as a Service, Service Oriented Architectures, Information Lifecycle Management, Agility, Records Management

INTRODUCTION

The notion of Software as a Service has been put forward as part of a continued quest to provide agile, flexible software to support rapidly changing business needs. (Bennett, Layzell et al. 2000). As part of loosely coupled Service Oriented Architectures, software can be delivered for use as needed (Ren and Lyytinen 2008). From the business perspective, this has the potential not only to support flexibility, but also to reduce capital costs and risk, as the entire technical infrastructure can be hosted remotely, and users only pay when they use the software. Many software services have already been built, and are being put together to support business processes successfully – for example software services produced by Salesforce are now used in multiple industries, and to cover multiple business processes (Salesforce.com 2009). SAP has delivered over 2800 services to date using their new, service oriented approach (Frear 2009).

Furthermore, services are being developed in the context of open source architectures, using agreed industry standards (Ren and Lyytinen 2008). Communities of users are being encouraged by suppliers to exchange services they have developed. Salesforce.com, for example, provides the platform force.com (Salesforce.com 2007), and Appexchange as a forum for the exchange of software services created by users (Salesforce.com 2009). SAP is repositioning itself in the market to include the provision of integration services, and, interestingly, brokering agreement on the semantics of data elements (Businessweek 2005; Frear 2009).

It is increasingly possible, therefore, that businesses will move to a model where the majority of their processes are supported by a composition of software services, from different providers, but using industry standards and service oriented architecture platforms for integration. Critical to this model is the idea that services have certain characteristics: they are loosely coupled: they are business oriented, have similar levels of granularity as the services with which they interact, have explicit boundaries, and are autonomous (Ren and Lyytinen 2008). They also exhibit late binding, in other words they are configured to meet a specific need at a specific time, executed, and then discarded. (Bennett, Layzell et al. 2000) p3.

Such characteristics are similar to those defined in object oriented methods. Indeed, a minority of services are integrated using the Common Object Requesting Broker Architecture (Ren and Lyytinen 2008). An issue with object oriented methods was that, because data was encapsulated in the object, it did not persist outside it. This remained a problem for many years – Grady Booch’s paper on object oriented development appeared in 1986, (Booch 1986) and yet the problem of persistence of data was still being addressed in research articles in the late 1990s – see for example (Keller and Wiederhold 1999).
For Software as a Service, the problem goes further. Persistence of data only means ensuring that data is available from one use of a service to another. This is not enough. There are many other reasons for keeping information, other than for their use in business processes. These include legislative and archival needs. This more comprehensive requirement is here termed data longevity and defined as follows:-

- The legal, business and other requirements for data to exist outside of, and for different time frames from, the software applications and services that produce or use it.

The contribution of this paper is to characterise data longevity requirements for Service Oriented Architectures, and investigate the extent to which those requirements are currently being met. Specifically, the following questions are asked:-

- What are the requirements for data longevity in Service Oriented Architectures?
- Do current Service Oriented Architectures currently meet those requirements?
- How should IS managers, seeking to use Service Oriented Architectures, ensure that data longevity is appropriately addressed?

The paper begins with a short definition and description of Software as a Service, and Service Oriented Architectures. The provenance of Service Oriented architectures is summarised and discussed in terms of the possible intrinsic problems that may be implied for managing long term information needs. A set of requirements for data longevity are then developed as follows. Firstly, the longevity needs of data used in systems which support business processes is investigated, using SAP as an example. Four requirements are identified as a result of this discussion. Those requirements are then examined further by applying them to more detailed data – in this case a subset of HR data as kept in SAP, and a fifth requirement is found as a consequence. Two areas of literature that could help define requirements for data longevity in Service Oriented Architectures, namely Records Management models (Bearman 1993; Gilliland-Swetland and Eppard 2000; Miller 2003; Evans, Reed et al. 2008; Reed 2008), and Information Lifecycle models (IBM 2005; SAP 2008; SNIA 2009) are then used to examine the five requirements and to identify two more requirements. Having identified a list of seven requirements for data longevity in Service Oriented Architectures, the literature on Service Oriented Architectures is reviewed to examine the extent to which those requirements are fulfilled. Gaps are identified, an approach to data longevity for managers seeking to use Service Oriented Architectures is suggested, and areas for further research work are discussed.

SOFTWARE AS A SERVICE AND SERVICE ORIENTED ARCHITECTURES

Software as a service describes a way of fulfilling a specific business support need by means of software which is configured to meet that need, executed and then discarded. The business pays for this software only when it uses it – as a service (Bennett, Layzell et al. 2000). The delivery of such services is by means of Service Oriented Architectures. There are several, different definitions of service oriented architecture, depending on whether business perspective or technical perspective is being considered. In general, however, service oriented architectures provide a framework for allowing distributed computing services to be integrated, identified and configured. Identification is on the basis of a business oriented definition, and hence systematic reuse at a business level becomes possible, allowing services to be assembled to provide support for business processes in a way which can adapt to business change. (Ren and Lyytinen 2008).

Increasingly, such frameworks incorporate design standards such as Service Oriented Analysis and Design (SOAD), and Service – Oriented Modelling and Architecture (SOMA) (Zimmerman, Krogdahl et al. 2004), meaning that they can be shared across different organisations. Service Oriented Architectures are not platform specific. However, the most popular means of delivery is now the web. In that context, some web based services use peer to peer architectures and so can only be used on a limited basis. Others, however, make use of a series of communication standards defined by W3C and can therefore be used broadly across organisations. (Ren and Lyytinen 2008).

There are three core principles to Service Oriented Architectures. Scalability means that services are recursively composed of smaller services (Bennett, Layzell et al. 2000) and depends on the loose coupling of services (Ren and Lyytinen 2008). Reusability occurs because services have a clear business meaning and explicit functional boundary, and so can be offered as reusable units to be assembled to support business processes (Ren and Lyytinen 2008). Agility is a result of the fact that services can be assembled as required to support changing business need (Ren and Lyytinen 2008).

Software as a Service and Service Oriented Architectures are already a reality. Salesforce.com, for example, introduced its first service offering on February 2000 (Salesforce.com 2009) p 1. Based originally on Customer Relationship Management applications, and still with a substantial offering of services in that area,
Salesforce.com has now expanded to include a platform as a service. Force.com. Customers can develop their own application on this platform, and can also sample, share and install services developed by other customers using Appexchange (Salesforce.com 2009) p9. Salesforce.com now has approximately 55,400 customers worldwide, and claims to be the leading provider of Software as a Service, based on revenues and market share (Salesforce.com 2009) p1. They have 3,566 employees and had a revenue of $US 100million in 2008/9.

Software as a Service is offered by several other vendors. For example, SAP offers over 2,800 services as part of its new, Service Oriented Architecture approach. Most of these are written in non proprietary code, such as Java, and integration with third party services is expected, and, indeed encouraged. (Frear 2009).

It should be noted that the provenance of Service Oriented Architectures implies possible intrinsic problems with managing long term information needs. Ren and Lytyinen (2008) point out that the development of Service Oriented Architectures has been influenced by previous software architectures designed to improve the way in which software could fulfil business needs. Those previous system architectures include object-oriented methods, client-server architectures, multi-agent systems and DCOM/CORBA. The predecessor architectures that they list had problems, never fully resolved, with how persistent data should be treated. Grady Booch’s original, seminal article on object oriented design (1986) used the example of a cruise-control system to illustrate “an approach to software design in which the decomposition of a system is based on the concept of an object [where] an object is an entity whose behavior is characterized by the actions that it suffers and that it requires of other objects” (211). A cruise control system, at least as envisaged in 1986, would not have required any of the data items (for example system on/off, engine on/off, pulses from wheel) to be kept for any length of time. By 1989, the Object Oriented Database System Manifesto had been created (Atkinson, Bancilhon et al. 1989) in recognition of the fact that, while object oriented database systems were being developed there was no “clear specification” or “strong theoretical framework”. Work on persistence and objects was still being researched at the end of the last century – see for example (Keller and Wiederhold 1999)

REQUIREMENTS FOR DATA LONGEVITY IN SYSTEMS SUPPORTING BUSINESS PROCESSES

The conceptualisation of data within Enterprise Systems is used as a starting point for the development of a series of requirements for data longevity. This is because Enterprise Systems have a well-discussed ability to integrate data across transactions (see for example (Davenport 1998). For the purposes of this paper, one Enterprise System is considered in detail, namely SAP (Calvert and SAP 2008; SAP 2009; SAP 2009a). As discussed later, further research could establish whether the conceptualisation of data used by other Enterprise Systems, for example Oracle (Oracle 2009) would confirm these requirements, or extend or change them. In SAP ERP 6.0, types of data, the way in which they are integrated, and the way in which they are recorded, can be summarised as follows:-

- **Transaction data**: data created when a business process is executed such as creating a sales order. Such data is held in documents such as sales orders. (SAP 2009) p 15. These documents are unique to the particular date and time of the transaction.
- **Master data**: which is used for several business processes, and kept longer term. Master data is integrated into individual processes, and hence individual transaction documents. For example, a sales order will use information from customer master data, and material (inventory) master data. (SAP 2009) p 15. Reports on master data are always available from the system, and an audit trail is kept by the system of any changes made to this data.
- **Organisational data**: This describes an organisation in terms of how it is structured to support all its business functions including sales, procurement, HR, etc.(SAP 2009) Organisational data is integrated into both transaction data and master data. For example, material (master) data will include the location where the material stock is kept (plant) (SAP 2009) p 348. If that material is part of a sales order (transaction), that order will also contain information about the sales organisation responsible for it, and the distribution channel (eg retail, wholesale, internet) by which it will be delivered (SAP 2009) p 347. Organisational elements can be displayed by the system, and an audit trail can demonstrate any changes to them.

Using this typology we can begin to explore the problems that might arise if support for a business process is offered as a service. A software service to support the creation of a sales order, and its recording as a document, would be unlikely in normal circumstances to require ongoing access to integrated records. However, there are particular circumstances where this access might be required. If a product is recalled, for example (IBM 2005), then access to customer data is required – if that data has subsequently been updated, then the updated record
will need to be accessed. Similarly, if organisational data has changed, both the old and new organisational elements will need to be known.

It should be noted that organisational data is relevant to data longevity from a different perspective: it helps to define responsibility for the data, in terms of who can Create it, Update it, and Read it (known as CRUD, although SAP almost never allows the Deletion of data). In SAP, this is done by the creation of views based on organisational elements, and the creation of access rights for individuals or groups of individuals within those organisational elements (SAP 2009).

Further complexity arises when data is being kept about human beings. In this case, data can be subject to two, often contradictory sets of legislation – those, such as the NSW State Records Act which stipulate the length of time for which information needs to remain publicly available (SRA 2005, 1998), and those, such as the UK Data Protection Act which stipulate what can be disclosed about a person, and what data must be accessible to the person to whom it pertains (DPA 1998).

Data about human beings, like the generic data discussed above can consist of an integration of transaction data, master data and organisational data. It is likely to need ongoing access to the integrated elements of which it consists. For example, if a person is hired within an organisation, then the hiring transaction must be recorded, along with the master data pertaining to the person’s details such as name and address, and organisational data in terms of where in the organisation the person has been hired. According to the NSW State Records Act, some elements of Human Resource records for all public sector employees need to be kept in perpetuity (SRA 2005, 1998). Human Resource data can be very complex – a subset of that data for SAP is given in Table 2 as an example (Seethamraju and SAP 2009). Organisational structures change, and certain roles and positions may cease to exist over time, to the extent that putting together appropriate, meaningful records for archiving can be difficult. Moreover, different parts of the record would be subject to different privacy requirements. As another example, University student information is kept as master data. The series of student enrolments culminating in the award or a degree can be seen as a series of transactions, involving both student and unit of study master data. The original enrolment in the degree is a permanent record, as is the resultant degree award. However each unit of study has the potential to change each year, making the detailed reconstruction of the record difficult if students need, for example, to indicate to an employer exactly what they studied in a particular unit of study.

The above discussion is summarised in Table 1 below:-

**Table 1: A taxonomy of data longevity requirements based on SAP ECC 6.0**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Transaction data</th>
<th>Master data</th>
<th>Organisational data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Where data is integrated from several sources, access to the <strong>changing</strong> integrated data is required.</td>
<td>Each transaction document needs access to relevant master and organisational data. In exceptional circumstances, such as product recall this can be the case. Where the “transactions” form part of the record for a human being, such as a student transcript, the need could be ongoing for the lifetime of the student.</td>
<td>Master data are created with reference to specific organisational elements. For master data which includes human access there may be a requirement for long term access to transaction data, eg a student’s graduation or a person’s recruitment into an organisation. There may also be a requirement for access to organisational data, eg in HR records, where a person has been employed.</td>
<td>Organisational elements are created with reference to other organisational elements. There may be a long term need to keep records of the activities undertaken by organisational elements.</td>
</tr>
<tr>
<td>2. Definition of Create, Read, Update, Delete responsibilities</td>
<td>Required where transactions are part of an ongoing record for a human being eg student transcript, medical record etc.</td>
<td>Required for lifetime of master data</td>
<td>Required for lifetime of organisational data.</td>
</tr>
<tr>
<td>3. Required to fulfil privacy legislation</td>
<td>For transactions involving human beings, eg medical, student, insurance.</td>
<td>For master data involving human beings</td>
<td>Where organisational data is implicit in data about human beings.</td>
</tr>
<tr>
<td>4. Required to</td>
<td>For transactions involving</td>
<td>For master data involving</td>
<td>Where organisational data is</td>
</tr>
</tbody>
</table>
fulfil public
availability
legislation
human beings eg
recruitment data may have
to be kept for time frames
defined in law.
human beings data may
have to be kept for time
frames defined in law.
implicit in data about human
beings it may have to be
kept for time frames defined
in law.

Table 1 gives us four requirements for data longevity:

1. Ongoing access to all sources used in integrating data, including access to changes within those sources.
2. Ongoing definition of Create, Read, Update and Delete responsibilities
3. Definition of the ongoing privacy requirements for each element within that data.
4. Definition of the ongoing need for each element of the data to be available as required by records legislation

In order to improve the completeness of this list of requirements, it is examined in two ways, below. Firstly, the issue of data related to human beings is explored in more detail using a specific example, based on a subset of the human resources (HR) data in SAP. Secondly, literature in related areas is explored, namely recording keeping (Gilliland-Swatland and Eppard 2000; Evans, Reed et al. 2008; Reed 2008; CRKM 2009) and information lifecycle management (SNIA 2004a; IBM 2005; SAP 2008)

A SPECIFIC EXAMPLE: HUMAN RESOURCE RECORDS IN SAP

As discussed above, Human Resources data may have complex requirements regarding longevity. Some examples of HR data are therefore used as a test of the completeness of the four requirements for data longevity identified above.

HR data in SAP ECC5.0 is kept in the form of information types, or infotypes for short. These infotypes include basic personal data, basic contract data, gross/net payroll, planning data and working times (Seethamraju and SAP 2009). Each infotype contains subtypes. Some elements from the subtypes of basic personal data are shown in Table 2. The discussion is limited to data longevity requirements 1, 3 and 4, as it is not possible to discuss requirement 2 (CRUD information) outside the context of a specific organisation.

This analysis identifies that the three requirements discussed are all relevant to HR records. Furthermore, it illustrates the differing needs for different subsets of data. For example, an individual’s historical bank details are unlikely to be required to be kept. These are available from other systems. There are, however, privacy issues with using them. By contrast, information on organisational assignment may require substantial long term histories of integrated data, but have little privacy issues. This analysis stresses the importance of identifying the exact subset of data used in each transaction, so that its longevity requirements are appropriately mapped. An investigation of the way in which availability legislation is met identifies a new requirement:

5. Where data can be obtained from several sources (eg name and address data) the nomination of one source as responsible for its upkeep. This concept is commonly known as the “single source of truth”.

Table 2: Contents of SAP-ECC5.0 infotype on basic personal data

<table>
<thead>
<tr>
<th>Subtype</th>
<th>Examples</th>
<th>1. Integration with other data</th>
<th>3. Privacy legislation</th>
<th>4. Availability legislation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actions</td>
<td>Hire</td>
<td>Links with organisational assignment. If organisation structures change there may be a need to keep information regarding the change.</td>
<td>Some actions may be only be appropriate for viewing by specific personnel. Some actions may be required to be made available to the person to whom they pertain.</td>
<td>Some actions such as hiring need to be incorporated in long term employment records required in legislation.</td>
</tr>
<tr>
<td>Organisational assignment</td>
<td>Whether the person is actively employed. Where in the enterprise structure the person is located, in what job and in which position described</td>
<td>Links with the overall organisational structure of the organisation, with job descriptions, and with positions based on those job descriptions. If any of these</td>
<td></td>
<td>Employment records likely to be required long term, including links to organisational elements which may no longer exist.</td>
</tr>
<tr>
<td>RECORDS MANAGEMENT AND INFORMATION LIFECYCLE MANAGEMENT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>In this section, the five longevity requirements identified above are discussed in terms of two bodies of relevant literature: records management and information lifecycle management.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Records management in the business process context has a long history. For example (Miller 2003) discusses German record keeping practice, which was first extensively used in the Middle Ages, as a post-hoc tool for keeping track of records. By the 18th century, the government administrations in the kingdom of Prussia had developed ways of locating and tracking files, and using journals to keep track of the process of handling of matters, hence transforming records management into a dynamic component of business processes. The challenges to record – keeping are, of course, far more complex in the electronic age. Gilliland Swetland and Eppard (2000) discuss specific requirements of modern record-keeping, and a large research project has recently investigated these in detail (Evans, Reed et al. 2008; Reed 2008; CRKM 2009).</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Information lifecycle management, by contrast, is a concept found mainly in the practitioner literature. The Storage Networking Industry Association (SNIA) defines it as follows: “Information Lifecycle Management is comprised of the policies, processes, practices and tools used to align the business value of information with the most appropriate and cost effective IT infrastructure from the time the information is conceived through to its final disposition. Information is aligned with business requirements through management policies and service levels associated with applications, metadata and data” (SNIA 2004a) p1.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
It should be noted that most current records management and information lifecycle management literature assume that the platforms on which data is currently held remain accessible for post hoc archiving and lifecycle management. This has specifically been recognised as a problem in record keeping (Evans, Reed et al. 2008). Information Lifecycle Management solutions assume the long term existence of databases, and applications which are persistent – see for example (SNIA 2004a; IBM 2005; SAP 2008),

Both the records management and Information Lifecycle Management literature are discussed below, in terms of longevity requirements that would arise in a service oriented architecture. Confirmation is found for some of the longevity requirements already discussed, and further needs are identified, as follows:-

**Identification of the boundaries of information.** This issue is identified within the record keeping literature. Gilliland Swetland and Eppard (2000) provide a definition of a record which is related to data longevity requirements 1, 2 and 4: namely, the ongoing integration of data, and the requirement that it fulfill legislative need: “Records are heterogeneous distributed objects comprising selected data elements that are pulled together by activity-related metadata such as audit trails, reports, and views through a process prescribed by the business function for a purpose that is juridically required. Identifying the boundaries of such intellectually complex objects and then moving those objects forward through time and through migrations without compromising their authentic status is a significant issue”. However, the notion of a boundary to integrated objects is not covered by the data longevity requirements discussed above, and is hence nominated as a new requirement:-

6. Where data is integrated from various sources, the definition of its boundaries for each of its longevity needs.

**The changing value and reliability of information over time** This concept is at the core of Information Lifecycle Management solutions – see for example (IBM 2005; SAP 2008). This includes not only the way in which the value of information changes over time, but the way in which that value can suddenly be reversed, for example in emergency situations where inactive patient records are suddenly required from healthcare systems, or sales records are required for product recalls. (IBM 2005). The records management literature also discusses the changing value of information, pointing out that records are temporally contingent, in that they take different values at different times, and are created for the purposes of time-bound actions. They also change in terms of reliability requirements – while they are active their reliability depends on procedural and technical control exercised during their creation and management. By contrast, if and when they becomes inactive then they are subject to archival management, with a focus on authenticity. (Gilliland-Swatland and Eppard 2000). The need for appropriate metadata to reflect these changes was explored within the Clever Record Keeping Project (CRKM 2009). That project identified a problem in that metadata is often applied retrospectively (Evans, Reed et al. 2008). This is unlikely to be possible in a Service Oriented architecture where there is no application as a whole to be examined retrospectively. This, then, leads to a seventh data longevity requirement:-

7. The ability to apply the metadata necessary for data longevity at the time the service is delivered.

The Clever Record Keeping project also identified the problem of ensuring that the ongoing locus of responsibility for record-keeping metadata is clear. This leads to a modification of data longevity requirement 2 to include metadata:-

2. Ongoing definition of Create, Read, Update and Delete responsibilities for both data and metadata.

**SERVICE ORIENTED ARCHITECTURES AND DATA LONGEVITY**

The above discussion, then has led to seven requirements being identified for fulfilling data longevity requirements in Service Oriented architectures.

1. Ongoing access to all sources used in integrating data, including access to changes within those sources.
2. Ongoing definition of Create, Read, Update and Delete responsibilities for both data and metadata
3. Definition of the ongoing privacy requirements for each element within that data.
4. Definition of the ongoing need for each element of the data to be available as required by records legislation
5. Where data can be obtained from several sources (eg name and address data) the nomination of one source as responsible for its upkeep. This concept is commonly known as the “single source of truth”.
6. Where data is integrated from various sources, the definition of its boundaries for each of its longevity needs.
7. The ability to apply the metadata necessary for data longevity at the time the service is delivered.
In this section, the ability for service oriented architectures to fulfil these seven requirements is discussed with reference to example commercial offerings. Two research prototypes are then discussed in terms of their contribution to data longevity requirements.

Current commercial offerings for Service Oriented Architectures appear to have a problem with the way in which persistent data is conceived and managed, particularly where services are integrated across different platforms. For example, a widely used service based platform produced by salesforce.com, to allow third parties to create and share services. It consists of the following components: application exchange, user interface as a service, logic as a service, integration as a service, database as a service, and global, trusted secure infrastructure. (Salesforce.com 2007). The database is provided at the time of need, and possibly persist for some time separate from the services using it. However, the only way that the seven requirements could be even partially addressed would be to ensure that the boundaries of all data that needed to be kept as an integrated record were within the Salesforce platform – that is, limited to one commercial vendor.

Similarly IBM provides a Smart SOA integration strategy for Enterprise Resource Planning Systems (ERPs). They provide an Enterprise Service Bus which allows vendor provided systems and tools to be combined in federated integration model. This model, however, does not explicitly include data – the data is part of the vendor provided systems, hence none of the seven requirements are explicitly met.

SAP’s vision of a service oriented architecture is based on a mixture of packaged implementation, bought services, services composed from components, and custom code. Once again, it is assumed that data is provided by means of the applications within the mix, and, as with IBM, the seven requirements are not specifically met.

Oracle explicitly include data in their service oriented architecture, and discuss the issues of reliability and availability of data. They provide a data grid to help ensure that issues are resolved. Once again, however, it is assumed that the data is held in a database as part of an application (Oracle 2007).

A model of inter-organisational service use is provided by the Software and Information Industry Association. They define the ASP (Application Service Provider) value chain, which consists of hardware, infrastructure software, services, applications, network service providers, all as suppliers in a chain to an ASP which in turn supplies a customer. No mention is made of data in this model. (SIAA 2001)

The commercial examples discussed here, then, would only be able to meet the requirements for data longevity if all the data was held on one platform. Even then, contractual arrangements would have to be made to ensure that all seven requirements were met. If the vision of services being integrated from multiple vendors becomes a reality (Bennett, Layzell et al. 2000; Ren and Lylytinen 2008), then there is no clear way in which these commercial examples could provide the necessary data longevity.

Turning to the research arena, two illustrative projects are discussed: The Clever Record Keeping project discussed earlier (Evans, Reed et al. 2008; CRKM 2009), and a recent prototype which deals with the complexities of extracting data from large integrated databases (Zhu, Turner et al. 2004).

The Clever Record Keeping research project ran from 2003 to 2006 at Monash University (CRKM 2009). The aim of the project was to investigate and prototype sustainable frameworks for the creating, capturing and managing metadata required for electronic records. Data or metadata from that available within business processes was included in the scope of the project. (Evans, Reed et al. 2008) p116. (quoting the work of, (Bearman 1993))

The project investigated the automation of the movement of recordkeeping metadata between business, records management and archival control systems with the aim of improving integration between and so improving the capacity to create and manage electronic records (Evans, Reed et al. 2008) p 117. Two prototypes were built to investigate the findings. The first prototype was a proof of concept, and found that there were problems in that metadata standards are not fully interoperable, and that some of the barriers to interoperability came from within the standards themselves. Furthermore, current environments tended to be silo based and see metadata schemas as a retrospective form of cataloguing. Their second approach took into account recent work on Service Oriented Architectures, and developed an architecture with a registry service, where metadata is assigned to objects, a translation service, and a validation service. (Evans, Reed et al. 2008). The need to more fully engage with service oriented architectures is put forward by one of the members of the Clever Record Keeping project. (Reed 2008) Once encouraging finding, in terms of Evan, Reed et al’s research, is the possibility that record-keeping can itself be delivered as a service (Evans, Reed et al. 2008; Reed 2008).

A project investigating the complexities of extracting data from large, integrated databases uses the UK’s National Health Service data as an example. It provides a model for ensuring that the information required for a specific patient, in a specific circumstance, can be integrated from multiple data sets. However, it assumes that
the data required for reconstruction are there, and are held according to policies which ensure they are accurate and complete. (Zhu, Turner et al. 2004) p3.

While further research would be needed to ensure the completeness of the review in this section, it seems likely that the seven requirements for data longevity are not currently fulfilled by commercial architectures. The Clever Record Keeping Project may provide some inroads into the problem, if its findings were further developed in the business process and service oriented arena. Zhu, Turner et al’s work (2004) may provide some help in reconstructing data to meet longevity needs, if it is extended to include service oriented concepts.

CURRENT SOLUTIONS FOR DATA LONGEVITY IN SOA

Three suggestions are put forward as to how IS managers should approach the issue of data longevity when contemplating a move to service oriented architectures. Firstly, they should examine their needs for data longevity using the seven requirements identified here as a starting point. Secondly, they should consider the specific service oriented architecture they intend to use. If they intend to use a hybrid model, which includes legacy systems and a persistent database, then they should consider whether that database is adequate to support all of their longevity needs. If they intend to use a purely service oriented architecture, then they may need to consider limiting themselves to a single vendor to provide those services. They should then use the seven longevity requirements as part of their contractual negotiations with vendors. They should always ensure that their architectures meet their information lifecycle management and records management needs as well as the need for flexibility and possible cost advantages that may be fulfilled by a service oriented architecture.

CONCLUSION

Service Oriented Architectures have the ability to radically improve the extent to which the requirements for flexibility in business process support can be met. A future is possible where services are drawn from a number of sources, integrated and run as needed, and then dispersed. However, businesses also need to keep some data for the long term. The contribution of this paper is to specify seven requirements for data longevity. These were developed by looking at an example Enterprise System, in overview and in detail, and then evaluating the requirements against those found in the information lifecycle and records management literature. Having established seven possible requirements, some example service oriented architectures were examined against the requirements. It is concluded that, at present, there may be issues with data longevity in service oriented architectures which would merit a further research programme as detailed below.

FURTHER RESEARCH

Several research streams are suggested in order to provide more robust conclusions regarding data longevity needs in service oriented architectures.

- Consolidation of data longevity requirements. In order to further identify these requirements, the SAP based investigation should be extended to other environments and to other business processes. This would include ensuring that all types of data relevant to the discussion have been represented.

- Further investigation of the literature on records management and information lifecycle management. This should include a detailed exploration of the way in which records management concepts are being developed in the business process arena. It should also include an exploration of information lifecycle management platforms, and the way they are being used in practice. Research prototypes in these areas should also be investigated.

- Detailed investigation of the way in which Service Oriented Architectures are being used in organisations, and how data longevity has been addressed in specific circumstances.

- Proposals for new architectural developments to reconcile data longevity needs with the agility requirements met by service oriented architectures.

REFERENCES


Salesforce.com (2007). Force.com: create and run any application on demand; platform as a service.


