Should ebXML be preferred when integrating e-government?

Jenny Backhouse, Associate Lecturer (j.backhouse@adfa.edu.au)
Tim Turner, Lecturer (t.turner@adfa.edu.au)
Irene Kuek (irenek@cs.adfa.edu.au)
School of Information Technology and Electrical Engineering
University of New South Wales @ Australian Defence Force Academy
Northcott Drive, Campbell, ACT, 2600, Australia
Contact: Jenny Backhouse +61 2 6268 8692

Abstract

The integration of e-government is the next major step in the evolution of internet technologies in government. The adoption of a comprehensive technology framework will ease development in this complex arena. This paper examines two of the main integration technologies available, ebXML and Web Services more generally. The paper briefly reviews e-government's progress to date and the evolution of the two technologies. A 'rule-of-thumb' heuristic is used to determine that ebXML is the preferred technology for integrating e-government.

Keywords

e-government, interoperability, ebXML, web services

1. Introduction

Governments all over the world that have articulated e-government programs or initiatives claim that integrated government services, across agencies and across levels of government, are a key part of delivering the benefits that e-government has for the citizenry. This is usually seen to result in some version of a 'one-stop' service approach where the citizen is able to transact some complex government business without having to deliberately navigate the labyrinthine bureaucracy that reflects ministerial portfolios and parliamentary fiduciary responsibility.

e-Government is seen as a key enabler of this potential because of the apparent 'plasticity' of information technology and information systems. Building systems that can work together is becoming more straightforward every day. The emergence of widely adopted inter-process communication using document and communications standards such as XML, SOAP and others, has raised the potential for greater cross-government integration.

In deciding which integration technology might best suit e-government, accommodation must be made of the differences between public and private sector organisations. The integration technologies we are considering arise with heavy private sector emphasis. Part of the consideration of the selection of a preferred alternative is which can better accommodate the unique requirements of the public sector.

There are (essentially) two camps that have risen to real prominence in the discussion of the potential for electronic integration of business systems, from which e-government integration derives. Importantly, these two camps have different actual

and philosophical roots. On one side are the proponents of ebXML which arose from a "standards first" approach. ebXML is a comprehensive, although largely theoretical, architecture for coordinating two or more systems in an integrated process to conclude realistic business transactions. On the other side are the proponents of Web Services which arose from individual vendors and, later, from vendor consortiums. Web Services is the growing approach to publishing internal system capabilities to external actors through XML-based adaptors. Real Web Services are appearing with increasing frequency, however the complexities of integrating these independently-created services into some coordinated whole is, as yet, poorly conceived, and untested.

This paper briefly reviews e-government, ebXML and Web Services developments before exploring in more detail how the benefits and inhibitors of these two alternative technology approaches might influence their adoption by governments seeking to implement integrated electronic government services. The paper concludes that for simple, low-risk activities, Web Services promises to allow 'quick wins', while complex, high-risk integrations are still probably best implemented in the more robust and theoretically sound ebXML.

2. A Brief Review of e-Government Progress

Governments around the world started using Internet technologies in 1990s. The adoption of these technologies was spurred by a series of major strategies released in different countries in response to the 'sudden' emergence of the World Wide Web into the public conscience. In Singapore in 1992, the government launched the IT2000 plan to bring Singapore into the forefront of information and communication technologies. In the United States in 1993, President Clinton focused government efforts on the National Information Infrastructure (Clinton, 1993). In Canada in 1993, a Blueprint for Renewing Government Services using Information Technology was issued (Office of Information Management Systems and Technology, 1993) that set the scene for substantial activity by the Canadian governments in e-government. In Australia, Prime Minister John Howard established the e-government agenda in his Investing for Growth statement (Department of Industry Science and Technology, 1997). In the United Kingdom, Prime Minister Tony Blair launched the Modernising Government agenda (Cabinet Office, 1999) that encapsulated his government's response to the opportunities of the information economy.

The principal focus of early activity in e-government was to establish a presence on the Internet. Web sites sprang up quickly but usually with little more than commonly available information and facts and figures about the agency hosting the site (Deloitte Research, 2000). There were some excellent and bold initial attempts at online service delivery in this period, in particular, Singapore's e-citizen portal, Hong Kong's ESD (electronic service delivery) and Australia's Business Entry Point. These initiatives offered online services that were the aggregation or combination of services offered by different agencies in the respective governments. Some were less automatic than appeared on the web, with facsimiles being produced and distributed behind the scenes in some instances; however, they gave a hint of what integrated e-government could mean. The rhetoric of various governments quickly adopted terms like "joined-up government", "seamless services", "one-stop online shop", and "anytime, anywhere."

The strategies mentioned earlier typically set a goal date by which 'all government services' would be offered online (or some similar target). In Australia it was 2001 (Department of Industry Science and Technology, 1997), in the United Kingdom it was 2005 (after an initial pitch at 2008 (Cabinet Office, 1999)). Although some governments have claimed the achievement of these goals (Alston, 2002), most of the electronic services available online fail to meet the high standards set by the rhetoric, especially when examined in the light of aims to provide seamless services that hide the machinery of government from the constituent.

Many governments are now explicitly refreshing their online strategies and developing plans to pursue this 'higher-hanging' fruit of integrated services. More collaboration is being planned, both across agencies in particular governments, across jurisdictions in countries with multiple levels of government, and between governments and the private sector (Cabinet Office, 1999; National Office for the Information Economy, 2002).

These aspirations are based on the idea that information technology (IT) can be integrated so that elements of necessary government administration conducted in different agencies (and jurisdictions) can be orchestrated to automatically pass data and decisions from one management system to another without the need for continuous interaction by the constituent.

The emergence of widely adopted inter-process communication using document and communications standards such as extended mark-up language (XML), simple object access protocol (SOAP) and others, has raised the potential for greater cross-government integration. These standards and others around them are being adopted by governments in the hope that the technological obstacles to truly seamless government will be overcome.

3. The Emergence of Web Services

The World Wide Web Consortium (W3C) defines a Web Service as "a software system designed to support interoperable machine-to-machine interaction over a network" (W3C, 2002). Web Services, as the name implies, are web-based and can be addressed by a Uniform Resource Locator (URL), thus allowing communication between web applications. The Web Services model means that "Web sites and Web-based applications no longer act as remote islands" (Butler Group, 2001a).

Using Web Services, a business can make an application or business process available over the Internet (Tsalgatidou & Pilioura, 2002). At the technical level, that application (or service) can be discovered and consumed by others dynamically thus negating the need for complex business integration initiatives or pre-determined agreements (Tsalgatidou & Pilioura, 2002). Such loose-coupling suits some e-business requirements. The service is effectively a 'black box' where the consumer of the service only needs to know the interface for invoking it, rather than the details of how that service has actually been implemented (Butler Group, 2001a). The interfaces to the service are themselves described in machine-addressable form (W3C, 2002).

One of the fundamental technologies for Web Services is XML (eXtensible Markup Language). The languages used to represent services, pass messages and describe data are XML-based. The cornerstone specifications for Web Services, along with XML, are:

- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
- Universal Description, Discovery and Integration (UDDI) (Roy & Ramanujan, 2001).

SOAP provides the interface by which the Web Service is invoked. It consists of rules for encoding application data in a content-neutral manner (Patil & Newcomber, 2003). SOAP has achieved wide acceptance in the marketplace (Alvord, 2002). WSDL is a language for describing the actual Web Services and the way in which they can be accessed. UDDI provides a repository where organisations can store and publish the Web Services they make available. Potential consumers of the Web Service then use UDDI to find services of relevance to them (uddi.org, n.d.).

More recently, specifications have been devised that extend Web Services to enable the co-ordination of higher-level business workflows. These specifications include:

- Business Process Execution Language for Web Services (BPEL4WS or simply BPEL)
- WS-Coordination
- WS-Transaction (Kreger, 2003).

These specifications aim to enable complex business workflows to be described and managed effectively. “The specifications will help organizations coordinate business processes and transactions within the enterprise and with partners and customers across heterogeneous systems” (xml.coverpages.org, 2003).

4. The Emergence of ebXML

ebXML (Electronic Business using eXtensible Markup Language) is “a modular suite of specifications that enables enterprises of any size and in any geographical location to conduct business over the Internet” (ebXML.org, n.d.). The development of this framework was undertaken, in 1999, by the Organization for the Advancement of Structured Information Standards (OASIS) and the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT). UN/CEFACT had the benefit of many years of experience in supporting Electronic Data Interchange (EDI), an early entrant in the field of electronic data interchange. The framework for ebXML was developed using a formal process that involved more than 2000 participants over an 18 month period (OASIS, 2003).

The motivation for the development of ebXML was the need for non-proprietary, low-cost standards to facilitate the emerging sphere of e-business (Kotok & Webber, 2001). Electronic Data Interchange (EDI) had been established for over 25 years and allowed businesses to interact electronically. However the complexity, lack of flexibility and cost of implementing EDI during that time was a barrier to its wide adoption, particularly for small to medium businesses (Levitt, 2001). Traditionally, EDI was largely implemented using relatively-expensive private networks. Although adopted in some key industries, such as the automotive industry, EDI never achieved

real critical mass across the enterprise spectrum (Kotok, 2003). Ideally, the use of ebXML would provide an e-business marketplace on a global scale, in which organisations of varying size could participate effectively (Kotok & Webber, 2001).

An additional motivation for the development of ebXML was to provide a unifying framework for e-business. The success of XML as a meta-language meant that a proliferation of XML schemas was being developed, many of them for e-business (Kotok, 2003).

The framework for ebXML focuses on the business-related issues of e-commerce (Patil & Newcomber, 2003). Consequently, the set of ebXML specifications addresses the broad spectrum of inter-business relations covering areas such as registries, business processes, data communication, collaboration agreements and messaging (ebXML.org, n.d.). As a framework, ebXML is not constrained to a specific XML schema. Rather, it provides the structure within which interaction can take place using the dialect of XML agreed on by the trading partners and relevant to their particular industry. There is work underway to develop government-specific XML dialects that would assist ebXML to support government operations (see for example US Government XML Working Group at http://www.xml.gov/working_group.asp).

A key component of ebXML is the registry, a publicly available resource. Businesses wishing to trade electronically register a Collaboration Protocol Profile (CPP) which includes the specifications of their business processes and business interfaces, such as trading documents (Butler Group, 2001b). Companies wishing to do business electronically search the registry to find potential partners whose trading requirements match their own and whose business processes are compatible. A Collaboration Protocol Agreement (CPA) can then be negotiated seamlessly between the companies detailing which business processes are being used in this instance. The CPA forms a contract between the two companies. Other specifications within ebXML, such as a business process schema and/or ebXML Messaging Specification would then typically be used in the implementation of the trading arrangements between the two companies.

Like Web Services, ebXML uses XML-based languages for tasks such as defining processes, data and collaboration profiles and SOAP for message packing. Similarly, it makes use of the underlying Internet and web protocols that provide interaction over the web.

So, with governments looking to increase their interaction and these two major technical approaches to achieving those goals, which one should governments adopt? We turn our attention to this question now. First, we examine each of the two technology approaches in more detail looking at their benefits and limitations. Second, with this more detailed (albeit still high-level) view, we turn to actually trying to choose.

5. The Benefits and Limitations of ebXML

As we have described, ebXML arose to facilitate automatic conduct of routine business between businesses in the electronic realm. The founding members of the

ebXML consortium knew that if real progress was to be made in promoting interoperability, common, open, standards would be required.

The ebXML consortium also understood that just defining data structures and interface standards would not be sufficient. The long history of EDI already provided useful document standards and evidence that companies can interact automatically in the right circumstances. However, just having the documents from EDI translated into XML to make them more web-compatible would be insufficient (Levitt, 2001). Businesses would continue to require the certainty of conventional business agreements and would need to be assured of this in the open, Internet environment in which ebXML was designed to operate. These requirements are critical for government too.

So, ebXML was designed from the outset as a complete architecture of interoperation between business partners (Aissi, Malu & Srinivasan, 2002). This comprehensive architectural form is where ebXML's greatest benefits and largest inhibitors lie.

To establish ebXML compliant interoperability a range of issues beyond just agreement on document formats is required. ebXML requires that the overall business process and contractual agreements are formalised (Aissi, Malu & Srinivasan, 2002; Kotok & Webber, 2001; Kreger, 2003). Through CPPs and CPAs, trading partners develop a level of certainty over transactions conducted electronically that are equivalent to non-electronic approaches (Aissi, Malu & Srinivasan, 2002). Once agreement is reached, the ebXML architecture provides a frame in which the specific process can be defined, including basic messaging standards and document formats (Chappell et al., 2001; Kotok & Webber, 2001).

This top-down and comprehensive approach lends substantial credibility, clarity and certainty to the future electronic interactions between partners (Chappell et al., 2001; Kotok & Webber, 2001). It also provides a definitive baseline should changes later be required or discovered.

However, this same comprehensiveness is ebXML's biggest inhibitor. The careful arrangements that are stipulated in the CPPs and CPAs and the way that security arrangements are spread through several elements of the architecture, lock participants into adopting the whole ebXML architecture (Aissi, Malu & Srinivasan, 2002). There is a substantial body of work to do before electronic trading can commence. Each transactional arrangement between partners requires some negotiation, even if only electronic, prior to the integration taking place. Similarly, changes to either party's approach to their element of the integration will almost certainly involve further negotiation.

Consequently, the top-down approach of ebXML best suits arrangements that will have a reasonable life-span (Chappell et al., 2001; Kotok & Webber, 2001). The initial discipline of establishing CPPs and then negotiating CPAs will require the certainty of some lengthy efficient operation to offset the initial costs (Chappell et al., 2001; Kotok & Webber, 2001). As ebXML integrations become increasingly common, this initial overhead will diminish, but it is a distinct disincentive to experimentation.

6. The Benefits and Limitations of Web Services

Web Services, since their introduction in 1999, have garnered widespread support from vendors at all levels, including some of the heavy-hitters in the software world. Both Microsoft and IBM have been active in producing products based on Web Services. Although not the first technology to enable business interactions over a network, Web Services have the advantage of being platform-independent, web-based and more flexible than other technologies for component-based development (Butler Group, 2001a; Vaughan-Nichols, 2002). Additionally, a Web Service offers the flexibility of greater granularity than a component in component-based development since a Web Service functions at the level of a business process.

XML, the meta-language on which Web Services specifications are based, has been widely adopted and tools for producing and processing it are readily available. The underlying Internet technologies used by Web Services, such as HTTP (HyperText Transfer Protocol), TCP/IP (Transmission Control Protocol/Internet Protocol) and FTP (File Transfer Protocol) are mature, non-proprietary and widely implemented.

Other than XML, the core specifications associated with Web Services, as mentioned earlier, are SOAP, WSDL and UDDI. These specifications are commonly accepted and real working implementations exist (Patil & Newcomber, 2003). Although these specifications were originally designed by particular groups of software vendors, they are now being supported by cross-vendor or standards-based bodies such as the World Wide Web Consortium (W3C) and OASIS.

One of the benefits offered by Web Services technology is the possibility of integrating back-end systems, including legacy systems, within the enterprise in a vendor-neutral manner. This potentially provides the organisation with a cost-effective integration solution as an alternative to the use of vendor-specific integration tools (Papazoglou, 2003).

The potential of Web Services for application integration is not restricted to intra-enterprise applications. The wider scope of dynamic business and/or government interaction obviously presents far greater opportunities but far greater challenges and risk. Consequently, most organizations have initially deployed Web Services internally, encircled by the protection of their organizational firewall. These implementations illustrate that Web Services offer real integration opportunities, but also point to the fact that inter-organisational integration is more than just connecting computer systems.

The majority of Web Services applications developed to date are of the request-response type. Essentially they cater for a simple business/government process where the interaction between the service and the service consumer is minimal and purely technical (Papazoglou, 2003). Typical examples of such services include requests for stock-price quotes or checks for credit-worthiness. While such services are useful, they do not tackle the more sophisticated electronic interactions a business needs to undertake in order to conduct serious e-business (Moore, 2003).

Real-world business transactions are characteristically more complex than the simple request-response model (Chappell et al., 2001; Kotok & Webber, 2001). A transaction

typically involves a sequence of interdependent requests and responses, occurring over varying periods of time, between the trading partners. A business process itself might involve more than one application and possibly span the enterprise boundary. For effective e-business, organizations need to be confident that these interactions will take place in the correct sequence and that mechanisms are in place to cope with any problems that may occur at various stages of the process (Chappell et al., 2001). The messaging interactions between the trading partners need to be both reliable and traceable. "Process oriented workflow systems and e-business applications require transactional support in order to orchestrate loosely coupled services into cohesive units of work and guarantee consistent and reliable execution" (Papazoglou, 2003).

As Jenz (2002) and others identify, the nature of business collaboration can be considered in three broad categories, requiring progressively greater collaboration from the businesses involved:

- Information services: provision of information such as stock quotes
- Integration services: provision of a service such as a reservation system
- Transaction services: a business process, such as product order and delivery, which involve a long-duration transaction

Current Web Services applications implement the lowest level of business collaboration (Roy & Ramanujan, 2001). Some of the key issues to solve in order to enable the use of Web services for more sophisticated collaboration include semantics, reliability and traceability, security and authentication.

As Jenz (2002) points out, while a Web Service is loosely-coupled from a technical point of view, from a organisational point of view, Web Services applications require tighter coupling between partners. "If anything changes that affects the availability of a Web Service, the interface of a Web Service or the semantics of a message, the service provider would need to proactively notify service requesters" (Jenz, 2002). This may not be a simple process since the Web Services model has no requirement for trading partners to know one another or to have any agreements in place.

The 'higher-level' Web Service specifications (BPEL4WS, WS-Coordination and WS-Transaction) that have been recently released are designed to address these types of concerns (Kreger, 2003; xml.coverpages.org, 2003). BPEL4WS, for example, is a flow language which can be used to define how activities can be linked in more complex business processes (Kreger, 2003; Weerawarana & Curbera, 2002). A current limitation of Web Services is the lack of maturity of these newer specifications and the extent to which overlapping specifications exist (Kreger, 2003; Roy & Ramanujan, 2001). "The abundance of overlapping standards for Web Services composition is overwhelming" (van der Aalst, 2003). As yet, there are few examples of this level of Web Services in real-world situations, although some tools are now starting to become available (Kotok, 2003).

The concept of Web Services is thus evolving from the initial, relatively straight-forward model. Essentially, the architecture for Web Services is being built from the bottom-up. Various bodies have contributed to different stages of this development including diverse vendor groupings, the W3C and OASIS. An important issue is how, and by whom, this evolution is being co-ordinated.

The W3C has recently devised a formal conceptual model for Web Services Architecture (WSA). This conceptual model is designed to support the more sophisticated tasks Web Services are being asked to perform and to ensure that the applications developed will be both inter-operable and extensible (W3C, 2002). A vendor grouping, including IBM and Microsoft, has formed the Web Services Interoperability Organization (WS-I) to "promote consistent and reliable interoperability among Web services, and articulate a common industry vision for Web services" (Microsoft, 2002) However, the tools and test suites provided by WS-I currently only relate to 'basic-level' Web services and not the higher-level specifications which are still the subject of much debate. "…The fulfillment of the Web Services promise to simplify integration depends on standards that today are the subject of rancorous debate by competing vendors and standards organizations" (Varon, 2003).

Web Services applications are being developed but currently, implementing Web Services enterprise-wide let alone inter-enterprise is neither simple not cheap. "… planning, deploying and managing an enterprise-wide Web Services implementation can be dauntingly complex" (McCarthy, 2003). Any large scale implementation inevitably requires guidance and tools from one vendor or another, potentially involving some proprietary solutions for specifications not yet fully adopted.

This lack of fully defined standards, along with the inevitable security issues that arise from a decentralized system operating in a heterogeneous environment, still present serious issues for organisations, both public and private, considering a Web Services implementation (Gandy, 2004).

7. Designing on-line services for public organisations

It is common rhetoric that e-business is good for business, and that this applies equally in the public and private sectors. Public and private sector organisations, however, are different in focus and operation. Below we summarise the differences between public and private organisations, identified by one author in earlier work (Turner, 2002), when considering the design of online services:

Nature of Service

- Coerciveness, or the unavoidable nature, of government services—public sector 'customers' are frequently compelled to adopt services, where private sector organisations must 'lure' customers to adopt
- Breadth of impact of those services— public sector organisations are usually targeted at the entire populace where private organisations can and do pick target markets

Public Scrutiny

- Interdependency between government agencies—increasingly, public sector services are delivered through the coordinated operations of several agencies; the private sector works in partnerships and collaborations too, but most commercial activities are carried out in competition rather than cooperation
- Accountability—it is important in both private and public organisations, but is generally more important in public organisations (Bozeman & Bretschneider, 1986)

The process-oriented nature of bureaucracy—aligned to accountability, processes in the public sector are usually more rigorous and less flexible than those in the private sector; hence the concept of red tape.

Unique Public Expectations

Privacy—the public generally expects the public sector to protect individual privacy, where private sector transactions will frequently trade-off privacy for service increase

Equity of access to services—aligned with breadth of access (above) the public sector is expected to make services available to all potential recipients even if not online, where the private sector chooses its range of access alternatives on the basis of commercial factors

Fee-free services—the public will frequently feel that public sector services should be free as they are delivered using tax-payer funds, where the private sector is inherently about recouping costs (and profit) through fees and charges.

Political cycles—the changes of government policy that occur at elections (even when the government does not change) will frequently change the nature, reach, or process of services, particularly if more than one jurisdiction is involved.

8. Criteria for Selection for Use in Integrated e-Government

Having set the organisational and technological scene we must finally build a heuristic by which to select one technology over another. A comprehensive selection approach with robust and validated criteria is outside the scope of this paper, but we offer here a 'rule-of-thumb' set with some justifications. Clearly, further research is needed for this element of our argument.

Our 'rule-of-thumb' heuristic involves four major criteria:

- **Technological sustainability**—governments tend to focus on mainstream approaches for IT to maximise effectiveness of tax dollar expenditure. Consequently, governments are typically adopting open standards and increasingly moving away from proprietary offerings, to the extent that they can still achieve their technological objectives.
- **Robust and secure**—technological failure or security breaches underpin failings in trust in government both at constituent level and between government partners in an integrated service offering. Security and constituent data privacy are probably THE issues in e-government at present and are emphasised when considering integrated e-government.
- **Accountable**—essentially, all government transactions are open to scrutiny. Integrated transactions are likely to be problematic because of multi-party involvement and the limitations placed on scrutineers by changes in jurisdiction and law within the one transaction sequence. Integrated services must facilitate scrutiny as much as possible within this broader legal and constitutional context.

- **Flexible**—Policy changes, especially when considering multiple jurisdictions are sufficiently frequent that any integrated service must be able to respond briskly and without 'breaking'.

We can now move to evaluating the two major integration technologies for their appropriateness for use in integrated e-government. We will outline the claims of each technology against our rule-of-thumb selection criteria and then summarise our findings and draw some conclusions.

8.1 Technological sustainability

Both technologies are roughly equivalent against this criterion. They both rely upon open standards, albeit standards that are still evolving, and are themselves supported by a substantial bulk of the information technology industry as 'standard' approaches. The small number of full ebXML implementations, in contrast to those using Web Services, may slightly favour Web Services as the preferred standard.

8.2 Robust and secure

ebXML is probably preferred here because of its more comprehensive approach. The explicit agreements established in CPAs and the definition of the overarching business process over the actual technological implementation allow for more robust integration between partners. ebXML has security for transactions designed into the architecture allowing for complete implementation of secure end-to-end transactions. Individual web services are arguably robust and can be made secure. However, the orchestration of a collection of web services into an integrated service offering is not yet subject to a standard (although standards are in development—eg, WS-Coordination) and so no reliable approach can be assured. Also, Web Services have already developed a reputation for being available at the whim of the owner. This could lead to unexpected failings in integrated services if certain web services were simply not available at some later time.

8.3 Accountable

ebXML is the preferred technology here. The discipline of CPAs and definition of business processes encourages a clear understanding of how the process will be operated and would facilitate the inclusion of appropriate audit elements for future scrutiny. In contrast, an integrated service based on Web Services is inherently a loosely-coupled collection of black boxes offering certain defined outputs. Government accountability requirements are unlikely to be satisfied by black box approaches and many web service owners will be reluctant to publish specifics of their web services, particularly if the owner is in the private sector.

8.4 Flexible

The Web Services Architecture is likely to be preferred against this criterion. It's inherently loosely-coupled nature allows services that might have met previous requirements to relatively easily be swapped for new services that meet new requirements without substantial impact on other elements of the overall integrated service. In contrast, ebXML's 'overheads' of agreements and business process

definitions are likely to inhibit quick responses to changes in requirements for the integrated service.

Summarising these assessments (see Table 1), we believe that ebXML is a preferred technology for integrated e-government services. The emphasis on developing agreements on how the service will operate and then formally defining the process prior to implementing using open standards is likely to best suit government's needs. However, its relative inflexibility tends to indicate that integrations in areas where there is less volatility in policy and legislation would be ideal. Services surrounding identity verification across jurisdictions, welfare entitlement confirmations, and taxation are areas that might be particularly suited to 'full-scale' ebXML implementations.

Web Services, on the other hand, are also an important technology for government. They will almost certainly form the basis of many online services, particularly those that address relatively simple requirements and to offer new services quickly in response to rapidly changing social circumstances (eg, evolving emergency situations). Their very flexibility and quickness-to-market are the principal inhibitor for their broader adoption in large-scale integrations. The government arena requires a higher level of reliability and accountability than Web Services, at least in their current form, offer.

Criteria	WebServices	ebXML
Technological sustainability	• Relies on open standards • Many real-world implementations	• Relies on open standards • Fewer full implementations
Robust and secure	• Individual web services robust • Integrated web services not yet secure or standardised	• More comprehensive approach • Security designed in
Accountable	• Services are black-boxes, loosely-coupled • A likely problem for accountability	• Business processes defined beforehand • Roles clearly understood by partners • Potential for auditing
Flexible	• Loose-coupling allows greater flexibility	• Overhead of prior agreements make quick changes more difficult

Table 1: Summary of heuristic assessment of Web Services and ebXML

9. Conclusion

The government 'industry' is unique in that, to a large extent, the players in the industry are not competing. This allows collaboration between agencies and integration of their systems and services to be developed in a more considered

fashion. We have briefly reviewed the circumstances that now lead governments worldwide to be considering this important next stage in e-government and some of the technologies that are likely to play a definitive role in that integration. We have proposed a rule-of-thumb heuristic against which to contrast the capabilities of the two main integration technologies and in the light of that comparison, favour ebXML for large-scale integrations across governments and between governments. Web Services remain an important technology that government must exploit but currently lack the necessary discipline to underpin integrated services that must be robust, secure and accountable.

10. References

Aissi, S., Malu, P. & Srinivasan, K. 2002, 'E-Business Process Modelling: The Next Big Step', *Computer*, vol. 35, no. 5, pp. 55 - 62.

Alston, R. 2002, *Government Online a Success as Australia Moves to a New Era of eGovernment*, Available: [http://www.dcita.gov.au/Article/0,,0_1-2_15-4_103828,00.html] (23 Aug 02).

Alvord, G. 2002, *SOAP and Standardization*, Available: [http://www.gogallagher.com/about/whitepaper02.htm] (March 2004).

Bozeman, B. & Bretschneider, S. 1986, 'Public Management Information Systems: Theory and Prescription', *Public Administration Review*, vol. 46, pp. 475 - 487.

Butler Group 2001a, *Getting Wise to Web Services*, Available: [http://www.butlergroup.com] (February 2004).

Butler Group 2001b, *Working with ebXML*, Available: [http://www.butlergroup.com] (February 2004).

Cabinet Office 1999, *Modernising Government*, The Stationery Office, London.

Chappell, D. A., Chopra, V., Dubray, J.-J., van der Eijk, P., Evans, C., Harvey, B., McGrath, T., Nickull, D., Noordzij, M., Peat, B. & Vegt, J. 2001, *Professional ebXML Foundations*, Wrox Press, Birmingham, UK.

Clinton, W. J. 1993, 'Executive Order 93-09-15', In *United States Advisory Council on the National Information Infrastructure*, The White House, Washington.

Deloitte Research 2000, *At the Dawn of e-Government: The Citizen as Customer*, Available: [http://www.dc.com/obx/script.php?Name=getFile&reportname=at_the_dawn_of_egovernment.pdf&type=pdf
[Requires Free Registration]] (16 Jan 02).

Department of Industry Science and Technology 1997, *Investing for Growth*, Commonwealth of Australia, Canberra.

ebXML.org n.d., *About ebXML*, Available: [http://www.ebxml.org/geninfo.htm] (March, 2004).

Gandy, P. 2004, *Managing Web services boom is essential*, Available: [http://www.butlergroup.com] (January 2004).

Jenz, D. E. 2002, *ebXML and Web Services - Friends or Foes?*, Available: [http://www.webservices.org/index.php/article/articleview/451/] (October, 2003).

Kotok, A. 2003, *Getting Web Services Ready for Business*, Available: [http://www.webservices.org/index.php/article/articleview/1014/1/24/] (Feb 2004).

Kotok, A. & Webber, D. R. R. 2001, *ebXML: The New Global Standard for Doing Business Over the Internet*, New Riders, Indianapolis, USA.
Kreger, H. 2003, 'Fulfilling the Web Services Promise', *Communications of the ACM*, vol. 46, no. 6, pp. 29 - 34.
Levitt, J. 2001, *From EDI To XML and UDDI: A Brief History Of Web Services*, Available: [http://www.informationweek.com/story/IWK20010928S0006] (March 2004).
McCarthy, J. 2003, *Web Services on a Platter*, Available: [http://www.infoage.idg.com.au/pp.php?id=1103740516] (February 2004).
Microsoft 2002, *Industry Leaders Align Around Web Services Interoperability*, Available: [http://www.microsoft.com/presspass/press/2002/feb02/02-06InteropOrgPR.asp] (January 2004).
Moore, J. 2003, *Taming Web Services: Emerging technologies may bring order to XML-based integration*, Available: [http://www.fcw.com/fcw/articles/2003/0324/cov-web-03-24-03.asp] (March 2004).
National Office for the Information Economy 2002, *Better Services, Better Government*, Commonwealth of Australia.
OASIS 2003, *ebXML News, Implementations, Developments and more*, Available: [http://www.ebxml.org/ebxml_jmt/documents/ebxml_jan-01_2003.ppt] (Jan 2004).
Office of Information Management Systems and Technology 1993, *Blueprint for Renewing Government Services using Information Technology*, Government of Canada, Ottawa.
Papazoglou, M. 2003, 'Web Services and Business Transactions', *World Wide Web: Internet and Web Information Systems*, vol. 6, pp. 49-91.
Patil, S. & Newcomber, E. 2003, 'ebXML and Web Services', *DS Online ISSN: 1541-4922*, vol. 7, no. May/June.
Roy, J. & Ramanujan, A. 2001, 'Understanding Web Services', *IT Professional*, vol. 3, no. 6, pp. 69 - 73.
Tsalgatidou, A. & Pilioura, T. 2002, 'An Overview of Standards and Related Technology', *Distributed and Parallel Databases*, no. 12, pp. 135-162.
Turner, T. L. 2002, 'What are the implications of being a Public Organisation on designing Online Services?' in *IPAA National Conference*, Institute of Public Administration, Australia, Adelaide.
uddi.org n.d., *About UDDI*, Available: [http://uddi.org/about.html] (March 2004).
van der Aalst, W. M. P. 2003, 'Web Services - been there done that? Don't go with the flow: Web services composition standards exposed', *IEEE Intelligent Systems*, vol. 18, no. Jan/Feb.
Varon, E. 2003, *Calculated Risks*, Available: [http://www.cio.com.au/index.php?id=341175082] (March 2004).
Vaughan-Nichols, S. J. 2002, 'Web Services: Beyond the Hype', *Computer*, vol. 35, no. 2, pp. 18 - 21.
W3C 2002, *Web Services Architecture*, Available: [http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/#whatis] (March 2004).
Weerawarana, S. & Curbera, F. 2002, *Business Process with BPEL4Ws: Understanding BPEL4WS, Part 1*, Available: [http://www-106.ibm.com/developerworks/webservices/library/ws-bpelcol1/] (March 2004).

xml.coverpages.org 2003, *Technology Report: Business Process Execution Language for Web Services (BPEL4WS)*, Available: [http://xml.coverpages.org/bpel4ws.html] (March 2004).