Preface

INTRODUCTION

Since their origins in the 1990s as purpose built software applications catering for individual on-line users, portals have taken a central place in the web landscape of the 21st century. This is the second book in a series describing the evolving character of the Portal, and in particular its implementation using another typically 21st century technology, Service Oriented Architecture (SOA). Since the release of the first collection of writings on the subject, New Generation of Portal Software and Engineering: Emerging Technologies, the trends we wrote about then have become even clearer. The promise of SOA to simplify information technology development in a cost effective way, and the expectation of individuals that they would be able to access information tailored to them via their phone, laptop, or increasingly rarely their desktop, continue to drive portal technology adoption forward.

This second volume explains and reflects this trend in detail. As in the first it continues the approach of combining academic research and industry experience. The chapters here were originally published in an earlier format in the quarterly International Journal of Web Portals (IJWP). The approach we have followed in IJWP, and described in the preface to the first book in this series, remains true: ‘First, a strong understanding of Portals, SOA, and the published research in these fields. In these areas IJWP sought to build on previous research. Second was an enterprise-based experience of factors that challenge implementation of Portal and SOA projects in practice. This brought in not just the practical challenges of such a project, but an enterprise customer view of the customer-vendor relationship in a field which requires large investment by both customer and vendor, and in its current phase a risk about the future of Portals and SOA shared by both customers and vendors. By combining these perspectives, IJWP provided a unique approach to research in the field.’

By 2010, when the originals of the chapters here were first published, the impact of the Global Financial Crisis on long-term technology investment was entrenched. Understanding the benefits of a Portal is easy. But for many reasons, quantifying them at the outset of a project is difficult. This has often been the case in the introduction of new technologies. The introduction of the US system of federal highways is an often quoted example. It is impossible to doubt the commercial benefit which has been achieved in the past half century from this project. But to have asked planners in the 1950s to identify what uses would be made of the road system even two decades later would have been a silly. Portals share the same characteristics of infrastructure projects. They provide the basis for services and applications that cannot be imagined, and certainly not quantified, at the outset. Those who build them expect to reap the rewards. However, ‘I don’t know what it will be used for’ doesn’t sit well in a business case. Even, ‘Everyone else is doing it and we can’t afford to miss out’ sounds weak at a time when investments can
only be made if they promise significant return in a short timeframe such as 18 months. This can lead to a gulf between investment need and investment appetite.

Where practical evidence is required, this calls for practical research. The chapters in this book lend themselves to this purpose. Practitioners describing and reflecting on their experiences of practical challenges, and theoreticians looking at the next generation of purposes and approaches in the use of Portals. We hope you find this a useful approach, and that it assists you in meeting the challenge of determining next steps in an ever-changing technical and business environment.

In this volume we have once again grouped contributions by topic, rather than chronologically, as follows:

- **Portal Technology section:** new developments in Portals, featuring, in particular, IBM extensive research in SOA, cloud middleware and portal search tools
- **Section on Security, Architecture and Mobility in Portals** contains several chapters dedicated to the research in portal security, portal architectures and portal mobile clients
- **Practical experiences of business today:** a review of the experiences of users of portal and SOA technology.
- **Learning for future implementations:** experiences today that assist us in contributing to the success of future implementations.

This book is structured around these four areas, and each of these is now examined in detail.

**PORTAL TECHNOLOGY**

Today the Web is used as a means to enable people and business access to information, services, and to execute financial transactions. Businesses need the worldwide visibility in their respective marketplaces. They also have to provide reliable provision of e-services to customers in order to maintain be successful. The Internet has become an important delivery mechanism of business visibility. Web portals with well designed services significantly extend businesses capabilities to sell and buy worldwide. The company website and its useability plays an important role in maintaining and extending business opportunities over the Internet.

What is a portal and a portlet? Different rendering and selection mechanisms are required for different kinds of information or applications, but all of them rely on the portal’s infrastructure and operate on data or resources owned by the portal, like user profile information, persistent storage or access to managed content. Consequently, most of today’s portal implementations provide a component model that allows plugging components referred to as Portlets into the portal infrastructure. Portlets are user-facing, interactive web application components rendering markup fragments to be aggregated and displayed by the portal.

There are a number of key conditions which make the web experience for all users as if it were custom fit for them incorporating their preferences, devices, location, social networks, and behaviours. Businesses need to meaningfully interact with, and listen to customers. On the other hand, customers must transform their online experience to two-way information sharing. This means that integration and service provision must be easy. To create interactive, context-aware Web applications, the application
must be able to easily leverage and extend existing data sources such as CRM systems, social media sites, and back-end applications, as well as cloud-based services. However, developing application services through the traditional development cycle is a labour intensive and error prone process.

Content Management Systems (CMS) provide the way of achieving visibility and maintaining currency of content. A brief discussion of CMS by J Polgar (“Do You Need a Content Management System?”) provides a perspective on CMS design development issues. Usually the CMS software provides authoring tools designed to allow users with little or no knowledge of programming languages or markup languages to create and manage content with relative ease of use. These tools represent an advantage as the development cost can be low and the content can easily be maintained by the users. However, the customization of content presentation is often required. A typical CMS would provide a presentation layer displaying the content to regular Web-site visitors based on a set of templates. The templates are sometimes XLST files. The content and presentation designer can opt for a web site created fully by the custom code, use CMS with embedded custom code, or keep CMS and portal code separately. Some CMS vendors have developed highly complex products that are often too complex for smaller organizations, and the design and management of such web sites can be a very frustrating task.

Vendors often deliver so called cloud applications (also known as software as a service, or SaaS) that actually are only one of the models of cloud applications. Real cloud applications are capable of providing benefits if they are designed to be cloud applications, and delivered in a cloud model. In general, cloud application are characterized among others by multi-tenancy in cloud space, seamless integration on demand including business driven configurability, fast deployment, provision of full control to the owner organization, and supports application scaling. The cloud provider is responsible for maintaining sustainable IT infrastructure as well as negotiated SLAs.

The chapter by Jun-Jang, et. al titled ‘A Cloud Portal Architecture for Large Scale Application Service’ provides the cloud development framework called Cogito-C. The framework deals with application services development within the context of large corporate applications with thousands of application services being built and delivered. The framework enables real cloud application services to be developed and delivered. This framework is not using cloud simply as a better runtime engine. Rather, it is used as the development platform to accelerate and optimize the solution development process based on large scale application services.

One of the important features of portal application services is self service. Self service is utilized in many Human Resources applications such as SAP, Sage Software and IBM Lotus Workforce Management. In majority of applications, the self-service capabilities are designed as ‘out-of-the-box’ service with only a few options to customize. The chapter by O’Connor, et. al. titled ‘Lotus Workforce Management’ discusses the approach taken by Lotus Workforce Management software to HR self-service solution. The application focuses on providing three key features that allow organizations more choice and control over the implementation of a self-service solution. These features are extensibility, customization, and ease of integration which are implemented with one of the widely used portal engines such as IBM’s WebSphere Portal, currently v7. Extensibility is provided through the WebSphere Portal framework that lets users add or remove components and functionality and determine the structure of communication between portal resources. Integration with IBM WebSphere Portlet Factory gives developers the ability to customize and design a solution that is tailored to the user’s needs. WebSphere Portlet Factory is a very powerful and flexible tool for fast portlet building. It sits on top of a Service Oriented Architecture (SOA) and developers can easily use and deploy core assets and automatically assemble them into custom portlets.
Recently, we witnessed appearance of new portal standards such as WSRP 2.0 and JSR 286. Both standards greatly contributed to the portal capabilities. For example, inter-portlet communication in JSR 168 could only be achieved with great effort. Typical implementations would include vendor-specific extensions placed on top of the portlet. This solution often resulted in breaking inter-operability. Some solutions focused on portlets using a shared store to exchange data, such as the session context or a database. All of these workarounds typically required unstructured effort. Furthermore, the portals render and aggregate information into composite pages to provide information to users in a compact and easily consumed form. Among typical sources of information are web services. Traditional data-oriented web services require aggregating applications and provision of specific presentation logic for each of these web services. This approach is not suitable for dynamic integration of multiple business applications and content without using integration middleware such as Enterprise Service Bus (ESB) or similar middleware.

The chapter by J Polgar titled ‘Using WSRP 2.0 with JSR 168 and 286 Portlets’ examines the relationship of WSRP specification with the portlet specification JSR 168 and evaluates some shortcomings of WSRP specification 1.0. The conclusion postulates that a clear architectural approach combining usage of WSRP and AJAX is required to enable creation of standards based, customizable, and dynamically generated reusable portlets that have required interactivity, response time and usability. The chapter also discusses the principles of building web services using the Web Services for Remote Portlets (WSRP) specification. The specification builds on current standard technologies, such as WSDL (Web Services Definition Language), UDDI (Universal Description, Discovery and Integration), and SOAP (Simple Object Access Protocol). It aims to solve the problem of traditional data oriented web services which required the applications to be aggregated before any specific presentation logic could be applied for presenting the content. The portlet standard (JSR 168) complements the WSRP mechanism by defining a common platform and APIs for developing a UI in the form of portlets. WSRP enables reuse of an entire user interface. One of the advantages is that only one generic proxy is required to establish the connection.

The demand for integration using traditional (WSDL) web services in a portal forces other business partners to connect to these services in the Service Oriented Architecture (SOA) fashion. Such web services have to be published in the repository accessible to all partners such as UDDI. In addition, all published services would have to be maintained by business partners including their presentation logic. Web Services for Remote Portlets (WSRP) attempt to provide a solution for implementation of lightweight Service Oriented Architecture (SOA). The UDDI extension for WSRP enables discovery and access to user facing web services provided by business partners while eliminating the need to design user facing portlets locally. Most importantly, the remote portlets can be updated by web service providers from their own servers. Remote portlet Consumers are not required to make any changes in their portals to accommodate updated remote portlets. This approach results in easier maintenance, administration, low cost development and usage of shared resources. Furthermore, with the growing interest in SOA, WSRP should cooperate with ESB.

The chapter by T Polgar titled ‘WSRP, SOA and UDDI’ deals with the technical underpinning of the UDDI extensions for WSRP and their role in web service sharing among business partners. A brief description of the architectural view of using WSRP in enterprise integration tasks and the role Enterprise Service Bus (ESB) is presented to outline the importance of remote portlets in the integration process. Leveraging web services through portals by means of the Java Portlet and WSRP standards gives companies a relatively easy way to begin implementing an SOA. Most portals have built-in support for the
Java Portlet API and WSRP in the Portal Server which makes implementing a portal-based SOA even easier and cheaper. Portal support for the WSRP standard allows companies to easily create and offer SOA-style services and publish them in order to be accessed by other Consumers. The Consumers can combine several of these user facing services from diverse sources and portals to form the visual equivalent of composite applications. This approach delivers entire services to the other Consumer in a fashion which enables them to conveniently consume the services and use them without any programming effort. Furthermore, the Enterprise Service Bus (ESB) can be used to create a controlled messaging environment, thus enabling lightweight connectivity. Using WSRP and UDDI extensions for remote portlets makes the end-user completely shielded from the technical details of WSRP. In contrast to the standard use of data-oriented web services, any changes to the web service structure are implemented within the remote portlet and the Consumer is not affected by these changes.

Web analytics are typically branded as a tool for measuring website traffic. They can be equally well used as a tool for business research, and to measure the results of advertising campaigns and market research. Web analytics provides data on the number of visitors, page views, measure a visitor’s navigation through a website, and so on. This collection of data is typically compared against some metrics to indicate whether the web site is delivering expected values, and what improvements should be considered. These metrics are also used to improve a web site or marketing campaign’s audience response.

The use of web analytics in portal applications is discussed in chapter by J Polgar titled ‘Use of Web Analytics in Portals’. Tracking portal visits is important in order to obtain better understanding of which parts of the portal are delivering value. However, portals have unique attributes associated with the page composition techniques, page and portlet refresh. Portals always present multiple topics on the same page which pose specific challenges to explore exciting opportunities allowing the web designer to gain insights about portal usage and user behaviour. Furthermore, portals are inherently multidimensional, and an effective tool to monitor and analyse portal data usage must be able to support multidimensional analysis.

Web analytics or site analytics are used to provide data about the number of visitors, page views, show the traffic and popularity trends. Portals are inherently multidimensional web sites. In portal applications, the key to knowing what to track and monitor is understanding how the site is built and how the page URL is formed. In addition, portals are often used in conjunction with Content Management Systems (CMS). The use of site metrics to capture and measure user activity primarily to understand end user needs, behaviours and site usability enable the designers to build better portals and better target the content. It is often expected that a knowledge of user behaviour would lead to increases in revenue with better content targeting and can also impact the cost of automatic tuning. Site Analytics are also know as being a factor in reduction of testing costs with better designs.

In portals, the integration with site analyser tools is often performed by generating reports based on the portal site analyser logs or manually embedding tags into portlets and themes. A well designed portal is expected to provide an environment for the necessary collection of analytics data and offer seamless integration of the web analytics engine with the portal. Web Analytics are typically gathered in one of following ways: server-side log analysis, active page tagging, and click analytics.

In many well performing organizations, analytics has replaced intuition as the best way to answer questions about what markets to pursue, how to configure and price offerings, and how to identify where operations can be made more efficient in response to cost and environmental constraints. Yet, as much as business leaders are eager to capture the benefits of new intelligence, they need to take analytics the
full distance. Top performers are enacting their business analytics and optimization (BAO) vision, making it possible to make decisions operational and optimize business performance across the enterprise. To do this, they are using the most effective toolsets, governance and change management practices.

Modern web applications and servers such as a Portal require adequate support for integration of search services. The primary reasons are user focused information delivery and user interaction, as well as new technologies used to render such information for the user. Web crawlers in the past already had to deal with dynamic content and JavaScript generated content. The solution very often resulted in ignoring such web pages.

Portal Search supports the use of seedlists to make crawling websites and their metadata more efficient and to provide content owners fine-grained control over how content and metadata are crawled. WebSphere Portal provides a framework which propagates content and information through so-called ‘Seedlists’ - comparable to HTML based sitemaps, but richer in terms of features. Of course it mandates that information or content delivering applications need to be ‘search engine aware’ - it requires them to enable services and seedlists for fast, efficient and complete delivery of content and information. This would be the main integration point for search engines into the portal for Portal site search services with rich and user focused search experience. The chapter by Prokoph titled ‘Search Integration with WebSphere Portal: The Options and Challenges’ discusses the options of how such technologies can also allow for more efficient crawling of public Portal sites by the prominent Internet search engines as well as discussing some myths around search engine optimization. He states that it is obvious that it becomes very tedious for crawlers to focus on the core information of a ‘web page’. Ideally they should be able to dispose of any ornaments on such pages, like navigation bars, banners, and so on, and then focus on the core information provided on that page. So for any content published to such Portal pages, it would make sense to provide the crawler through a URL with that essential information only. Yet still it should be able to navigate the user via the search result list, to the correct context in which that specific content object is rendered. The Content Provider Framework provides the infrastructure for Seedlists. It defines for an entity within the Seedlist two types of URLs:

- crawler URL – as the name states, for a crawler to pick up the content itself, e.g. the content object from the WCM library which typically would get rendered through the Content Viewer portlet on one or more Portal pages
- display URL – this would be the URL that is given to a user to view that exact same content in the correct context of the Portal

In this way the search engine crawls and analyses the content delivered by the backend service through the crawler URL, whereas later on when searching, the user will be presented with the display URL in the search result list, to ensure that they see the information in the right context of the Portal.

As we already mentioned, portals are very good at aggregating and integrating applications so called at the glass on desktop PCs and laptop browsers. With the current trend of mobile devices, more and more users expect to access Portals on their mobile devices. The challenge to support multiple devices with different sizes of presentation medium is a difficult one. Current technology favours HTML for PCs and other desktops. However, standard HTML web pages cannot be delivered to most mobile devices. These devices have different capabilities such as screen sizes, image formats, input methods, etc. With thousands of devices in the marketplace and the frequent introduction of new devices, a Portal
cannot support the many types of mobile devices that want to connect to the Portal’s many applications. Fitzgerald and Van Landrum in their chapter titled ‘Challenges of Multi Device Support with Portals’ discuss the issues and solutions to this many-to-many relationship. Their solution is seen in the IBM Mobile Portal Accelerator which provides multiple device support from a Portal by using a version of XHTML called XDIME as the content markup and a multi-channel component coupled with a device repository to provide the proper device specific view. As a result, the page that is sent to the device is appropriate for that specific device and its capabilities, where no horizontal scrolling is required, all the information fits on the screen, the forms work, and all images are rendered properly creating a positive user experience.

SECURITY, ARCHITECTURE AND MOBILITY IN PORTALS

Computer security has become critical issue in entire IT industry. Many organizations are facing security threats both from employees and outside intruders. Web portals are not immune to hackers and on many occasions, hackers have broken the existing security barriers and have damaged the IT infrastructure. The chapter by Sultan and Kwan titled ‘Generalized Evidential Processing in Multiple Simultaneous Threat Detection in UNIX’ proposes a hybrid identity fusion model at decision level for Simultaneous Threat Detection systems. The hybrid model is comprised of mathematical and statistical data fusion engines: Dempster Shafer, Extended Dempster and Generalized Evidential Processing (GEP). The Simultaneous Threat Detection Systems improve threat detection rate by 39%. In terms of an efficiency & performance, the comparison of 3 inference engines of the Simultaneous Threat Detection Systems showed that GEP is the better data fusion model as compared to Dempster Shafer and Extended Dempster Shafer. GEP increased precision of threat detection from 56% to 95%.

Any producer of web-based material is interested in what users do with the pages they visit: what do they visit, how long do they spend there, and what do they do while there? In the educational domain, knowledge of a user’s activities can help to build a better educational experience. The intent is to build up a model of the user and to customise the site to desirable users. Current techniques for tracking behaviour of users in online learning systems have not been able to give deep views and enable user behaviour analysis. The chapter by Newmarch titled ‘Using Ajax to Track Student Attention’ shows how use of Ajax can provide a richer model of how users interact with Web systems. The chapter provides the description and results of case study deployed in one of the larger Melbourne teaching institutions. Ajax consists of a JavaScript call that can be made asynchronously to a web server. Typically such a request carries XML data, although this is not prescribed. The browser does not pause, or refresh while such a request is made. If a reply is received, the JavaScript engine may act on this. Some Ajax applications may use JavaScript to manipulate the browser’s DOM model, to cause apparently interactive responses. The advantage of Ajax is that it can avoid the fetch-wait-refresh cycle usual in following hyperlinks or submitting forms.

Average Revenue Per User (ARPU) is a measure of the revenue generated by Users of a particular business service. It is a term most commonly used by consumer communications and networking businesses. For mobile devices, they try to generate ARPU through network and content services (value-added services) that they make accessible to the User. It seems that the more accessible these services are, the greater the ARPU generated - the harder something is to find, the less likely someone is to use it.
The chapter by Young and Jessopp titled ‘How Thick Is Your Client?’ explores the potential continuum between ARPU and service discoverability for mobile services by comparing and contrasting various technologies with respect to development, user experience, security, and commercialisation. From the discussion presented in this chapter, it seems clear that increasing discoverability of services often involves more complex device integration efforts with the creation of a thin network hosted site accessed through the native device browser being the simplest but hardest to expose and thick native integration (idle screen, home screen) the most complex but highly surfaced. It is apparent that the more obvious the access method to a service is, the more likely a User is to make use of it at least once. The answer to the question, how thick is your client?, then appears to ideally involve facilitating the best User access possible to network services on a device-by-device basis. It is also to offer the User the choice of all possible access mechanisms (web, plug-in, portal, client) supported by their device; ‘horses for courses’ as it were. The consideration then is what the Business can financially justify to support this approach.

Software components composition can improve the efficiency of knowledge management by composing individual components together for a complex distributed application. There are two main research approaches in knowledge representation for component composition: the syntactic based approach and the semantic-based approach. The chapter by Khemakhem, et. al. titled ‘An Integration Ontology for Components Composition’ proposes an integrated ontology-supported software component composition, which provides a solution to knowledge management. The proposed search engine (SEC++) provides dual modes to perform component composition. Ontologies are employed to enrich semantics at both the component description and composition. SEC++ is an efficient search engine which help the developer to select components by considering two different contexts: single QoS-based component discovery and QoS-based optimization of component composition.

**PRACTICAL EXPERIENCES OF BUSINESS TODAY**

Technology functionality only goes part way to identifying benefits that organizations will achieve from investment in Portals and Service Oriented Architecture. A major theme of the International Journal of Web Portals has been look at the experience of organizations, whether they are corporations, government agencies, smaller enterprises, or the not-for-profit section. Has the promise been met? Have expected benefits eventuated?

The first contribution in this section asks a specific question, does an announcement of intention to implement a Portal affect the market valuation of a company? Gupta and Sharman in ‘Impact of Web Portal Announcements on Market Valuations’ identify that while significant research into the provision of electronic services has been undertaken, this relatively obvious metric has not been previously examined. Using the event-study methodology, they look at the impact of Portal announcements on a company’s share price, using a sample of 25 publicly traded companies. This looked at share price movement prior to the announcement, and then in the period after the announcement. The cases themselves include various factors including the size of the enterprise. It also looks at two different approaches to the use of Portals foreshadowed in publicly announced plans to create a Portal. The first of these is to expand the range of services that are currently offered to existing customers. In this, a company undertaking such an action is seen to be proactively addressing the changing environment. It can also be viewed as being able to adopt new technologies to assist existing services such as communication, collaboration, information
and personalization. The other area is to use the Portal to reach new customers, and a Portal provides a platform on which a company can move into other market segments. The findings of this report were that announcements of plans to implement a Portal provide a significant boost to market value.

The ubiquity of Portals for modern enterprises has in turn led to an expectation that technology service providers will have the necessary skills to advise and implement these solutions for their customers. The second chapter of this section, ‘Part of the Tool Kit: SOA and Good Business Practices’ is a case study conducted with Wong from the service provider, e-CentricInnovations. This company focuses on providing services to Fortune 500-type companies and government departments. It specializes in several technologies including SOA, Portals, and collaboration tools. At the time of publication, Web 2.0 technologies including social networking were just beginning to make their impact in the enterprise environment. Wong looks at the experience with Twitter, Facebook, Google and others. Since then these applications have gone on to change many social and corporate environments. This is similar to the way that e-mail had changed practices in the previous decade. The need for an approach to addressing technologies that previously hadn’t existed has therefore been born out. One irony described in this interview is the concern of many companies regarding allowing staff to share information via the company intranet. As Wong describes, in many cases the company’s de facto intranet has become Facebook. Over-tight internal controls have therefore resulted in a complete loss of company control over the channel. Social media have also exposed a generational divide among corporate executives. For the older ones, they don’t get it, but they don’t care because they will have retired in five or ten years. On the other hand, a small proportion are beginning to innovate. While the growth of social media has made corporate technology use more complicated, Service Oriented Architecture has the potential to make it more simple. For Wong, there is nothing in SOA that couldn’t have been done a decade earlier, with enough money, time, and a large enough team. However, SOA puts this functionality, in a standardized format, into the toolkit of every technology practitioner. It meets the long-held technology promise of reuse, ‘invent it once and use it many times’.

While the focus of much of the research described in this book has been for enterprise implementations, both corporate and government, many of the lessons are also applicable to the not-for-profit sector. A case study is provided based on an interview with Noble, an industry practitioner with extensive understanding of and experience in the sector. The key finding of this is that the sector, rather than being less demanding, is actually more demanding. This occurs because some of the challenges facing the sector are more exacting, while the resources available are significantly more limited. For example, in providing harm minimisation services to drug users, perceived privacy is an important feature in gaining the confidence of the client base. Challenges such as these place the not-for-profit sector at the forefront of demands for sophisticated Portal software, yet without the resource base to undertake this through a normal commercial path. While commercial software providers will generally assist by making their software available to the sector at a significant discount, when the sector seeks to access technical support, it is competing for services at the market rate. In contrast to this, a self-help approach between different organisations in the sector using open source software has been a common response. This shared approach has also resulted in the sector being a strong supporter of standardisation in Portal and SOA technologies.

In the last chapter in this section, ‘Portals, Technology and e-Learning’, Adamson looks at the benefits that Portals and Internet based technology generally can provide to e-learning. Some of the key attractions of e-learning are: the flexibility of delivery, in time, across geographies, across media formats; the rapid
LEARNING FOR FUTURE PORTAL AND SOA IMPLEMENTATIONS

The previous section examines the experiences of Portal and SOA implementation, and the impact that these have had. This section looks at possible approaches for improving such implementations. Richardson, a UK based practitioner with experience in major enterprise implementations, in ‘Improving Our Approach to Internet and SOA Projects’, describes the experience of implementing new technologies using existing tools poorly. His focus is on the project world, where he sees the dominant project management methodologies as a mixed blessing. While methodologies such as Prince2 have provided standardized approaches, the experience of projects based on these remain mixed. He counterposes this to focusing on both the defined project management methods, and the soft skills around people management. Where the hard skills of project management are reduced to a set of product features, and these are presented as a comprehensive approach to the requirements of complex project (the rule rather than the exception for Portal and SOA projects), failure awaits. If practitioners then blame the tools they have missed the point: the tools by themselves were never going to assure reliable project delivery.

The difficulty of learning from past Portal and Internet projects in improving the delivery of future projects is examined in ‘Challenges in Researching Portals and the Internet’ by Adamson from an historical perspective. Part of the challenge a decade after the dot-com crash of 200-01 is that with some $4 trillion share market value lost at that time, entire classes of business disappeared. It was impossible for even the best business model with the greatest governance and most competent staff to continue when the entire ecosystem in which they existed vanished. In these circumstances, instead of learning clear lessons from the first generation of e-businesses, we just learned that when a bubble bursts it isn’t good place to be. A second difficulty is that Portals cross many boundaries. Is the ability to recognize a customer and provide them with tailored services a marketing function, a service function, or a business development function? Each of these areas of a traditional organization could expect the Portal to be their responsibility. At the same time, wherever it ends up (and some large corporations have been known to establish multiple competing initiatives), that area will have a traditional skill set which will initially fail to appreciate the complexity and detail of the other functions they are now taking on. A third problem has been a misunderstanding of what stays the same and what changes with the Internet. Debates included whether technology mattered, whether traditional business theory was relevant, whether companies had to actually provide a good or service, and whether companies could indefinitely replace profitability with ‘first mover advantage’. Looking back many of these theories appear naïve. However, knowing which theories to keep and which to replace continues to be a challenge one decade after the crash. In addition, Portals and other e-business features have blurred the boundary between technology and business to an extraordinary extent. This has led to significant incorrect assumptions, as technology makes assumptions about business that are simply false, and vice versa. For example, the claim that technology
provides competitive advantage (as defined by Michael Porter) is incorrect: competitive advantage is achieved by the way technology is applied, not by the technology per se. While a technologist may call that splitting hairs, from a business investment perspective the difference is significant.

An extensive discussion about better ways to create Portals is provided by Lamantia from the Netherlands, in ‘Framework for Designing Portals’. The framework itself has been introduced in the previous volume in this series. In this volume we examine the elements of the framework, how these elements work together, and consider some large enterprises where this framework has been tested. These include the rules and relationships in regard to the basic structural elements of Containers and Connectors. While the comments in the previous paragraph describe the blurring of lines between business and technology from the perspective of understanding investment drivers, this framework considers another cause of blur: the proposal that technology simplification at the highest level will allow business users to directly make use of Portal technology as they wish. The complex side of technology (from programming to testing) will then be done ‘under the covers’. Lamantia’s proposal works seamlessly across the design framework, information architecture, Portal experience, portlets, technology management and governance, business design, and enterprise architecture.

The second of Lamantia’s chapters here looks at the goals that business will be pursuing as it engages more closely with what had previously been technical functions: collaboration, dialog, and support for social networking. The chapter looks at issues such as Portal management and governance. These are terms now shared extensively between business and technology, although it is difficult to determine whether there is a greater shared understanding of these terms than five or ten years ago. For example, on the simplest measure of technology engagement with the business, the relationship between a Chief Information Officer (CIO) and a Chief Operating Officer (COO) there is still no agreement. Strategists continue year after year to argue the merits of the CIO directly reporting to the Chief Executive Officer (CEO), versus the CIO reporting through the COO to the CEO. While this simple question arouses such difference of opinion, we cannot say that business and technology see eye-to-eye. The third of Lamantia’s chapters deals with practical experiences of large enterprises which have applied the framework, and the experiences gained from this. He concludes with an approach that combines the technical and business: ‘Looking around and ahead, we can see that the decentralized model underlying Web 2.0 reflects (or is driving, or both?) a fundamental structural shift; the information realm is ever more modular and granular. Consequently, the digital world is evolving complex structure at all levels of scale, and across all layers, from the organization of businesses into networks of operating units collaborating within and across corporate boundaries, to the structured data powering so many experiences. In fact, the whole digital / information realm - public, private, commercial, etc. - is rapidly coming to resemble the enterprise environments that encouraged the creation and use of the Building Blocks, and shaped their evolution as a design tool.’

CONCLUSION: IS OUR FUTURE IN THE CLOUDS?

In this book, there are several chapters discussing the difficulties and pitfalls the developers face when building web services, integrating applications, and architecting SOA frameworks. Clouds are next generation infrastructure which applies the mechanism of using virtualization technologies such as virtual machines. Three prime cloud delivery models are Infrastructure as Service (IaaS), Platform as Service
(PaaS), and Software as Service (SaaS). Cloud computing offers a pool of shared resources (applications, processors, storage and databases), on-demand usage of these resources in self-service fashion, elasticity (dynamic procurement), network access, and usage based metering.

It is expected that in the near future there will be millions of users using applications on the cloud. The reason seems to be in fast adoption and subsequent migration of client applications and processes to the new cloud service platform(s) currently being developed and delivered by major software and hardware companies such as IBM, Microsoft and others. It is not a dream to expect that in the future many larger software vendors could build their own cloud platforms and portfolios and sell cloud services. Such cloud platforms would have middleware to support service solutions, appropriate hardware and software for deploying customer cloud based applications, and usage metering services. The platform providers would provide consulting services to modify existing cloud models according to customer needs and administration services. Important administration services would cover management and tracking of business transactions, performance monitoring, data security, and many others currently burdening IT departments.

*What should we focus on when considering cloud as new home for company applications?*

**Horizontal capacity scaling and parallelism:** IT services and infrastructures always run out of capacity, and need to add capabilities on demand without investing in new infrastructure, training new personnel, or licensing new software. Cloud computing is typically seen as technology that uses the internet and central remote servers to maintain data and applications, and allows consumers and businesses to use applications without being involved in maintaining the IT infrastructure. In addition, it is believed that cloud resources allow applications scaling horizontally (scale up) with no capacity limits.

Typical applications are designed to be scaled vertically. However, the applications that are intended to be deployed on the cloud should be designed to scale-out (horizontally) rather than having the ability to scale-up. The process of scaling up is understood as adding more processing power by using faster CPUs, more RAM and larger throughput. All of that can be done by upgrading single server. But applications in the cloud need to have the ability to scale horizontally. It means adding more servers without any change in processing power. The design for horizontal scalability or parallel processing is the key to cloud computing architectures. The benefit of executing in parallel is that the same task can be completed faster using multiple servers. One of the key design principles is to ensure that the application is composed of loosely coupled processes, preferably based on SOA principles. This does not mean that the cloud enabled application would use a multi-threaded architecture (meaning resource sharing through mutexes works in monolithic applications). Clearly, multithreaded architecture does not provide any real advantage when there are multiple instances of the same application running on different servers.

*How do we maintain consistency of the shared resource across these instances when application is not design for utilizing parallelism?*

There are currently several suggestions as well as implementation in the research community. One method is using queues. The solution architects should aim at thread-safe the application which uses queues that cloud provides as means of sharing across instances. The application does not share resources any other way. However, queues are known for their negative impact on the performance of the system.
Many large applications use the ‘Memcached’ algorithm which is caching technology (http://memcached.org/, Brad Fitzpatrick (2004), Chris Bunch, Navraj Chohan, Chandra Krintz, Jovan Chohan, Jonathan Kupferman, Puneet Lakhina, Yiming Li, Yoshihide Nomura (2010)). This is a high-performance, distributed memory object caching system, which is intended for use in speeding up dynamic web applications by managing database load and session management. ‘Memcached’ allows configuring page snapshots at certain time intervals, avoiding the need to assemble together the same page over and over again thus saving some processing power of the underlying hardware. If a page is heavily based on DB reads with low sensitivity to time, the server load is reduced and the site becomes significantly more responsive. ‘Memcached’ is currently used on high-traffic sites such as Wikipedia and others.

Another method is to use the MapReduce algorithm (D. Thain, C. Moretti, and J. Hemmes. Chirp (2009), D. Thain, T. Tannenbaum, and M. Livny (2005), P. Pantel, E. Crestan, A. Borkovsky, A. Popescu, V. Vyas (2009), Brad Fitzpatrick (2004)) where the variables across instances are handled by ‘map,’ and the ‘reduce’ part handles the consistency across instances. MapReduce is a programming model and an associated implementation is typically used for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs, and a reduce function that merges all intermediate values which are associated with the same intermediate key. Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. In addition, grouping the results of like keys (i.e., gathering all the intermediate key/values for a given word) is handled by Apache Hadoop (http://hadoop.apache.org/) in the background. The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Delivering high-availability does not depend on hardware only, the library itself is designed to detect and handle failures at the application layer. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

As the solution for handling parallelism and distribution by applications is research in progress with many already successful implementations, there are still some areas where the cloud-enabled software is not readily available.

**IDE**: Applications have to be developed with parallelism in mind. It means there are some development platforms allowing the developers to write the code and test it. Cloud based applications have to be developed using appropriate IDEs. Ideally, these IDE would offer specific Cloud testing, performance testing, deployment options, plug-ins for multiple cloud providers, or embedded virtual cloud test environments.

**Middleware**: There is also important development in middleware. So far, middleware products are being used on dedicated physical servers. The advantage cloud utility model (pay-per-use) provides cannot be applied to applications and application platforms which are not designed to scale up or down based on SLAs. Therefore, a new generation of application servers, such as GigaSpaces XAP (http://www.gigaspaces.com/files/InsideXAP.pdf) and Appistry and their CloudIQ middleware, are gaining popularity among cloud users. GigaSpaces eXtreme Application Platform (XAP) is an application server with XAP middleware enabling to build scalable and highly performing enterprise applications in Java and .Net. Scalable, on-demand middleware is also an appealing solution for large enterprises which want to avoid bottlenecks by outsourcing parts of the middleware infrastructure into a SOA-Cloud. Appistry CloudIQ middleware simplified the process of deploying applications on to the cloud and between the clouds to simple drag and drop.
**Administration:** It is also envisaged that professional services focusing on system administration, configuration and network management will undergo significant innovation or more likely these services will have to be automated. The cloud-based middleware will provide administrative tools to manage space, distribution, and performance using an automated approach across multiple clouds.

All the above areas would have to be considered when designing applications living in the cloud. We would like to mention that this discussion does not cover many other issues associated with cloud computing such as security issues; this is just a brief peek into cloud computing landscape.

*Greg Adamson*  
*University of Melbourne, Australia*

*Jana Polgar*  
*Dialog IT, Australia*

**REFERENCES**


