Preface

A Portal provides a means of presenting information in an on-line environment. Portals today are widely associated with the World Wide Web, but the earliest Portals in the first half of the 1990s were either not based on the Web, or were ‘garden walled’, giving access to the Web but meeting most users’ needs within the proprietary environment. Service Oriented Architecture (SOA) and equivalent approaches to simplification of design and maximisation of reuse also have a long history within computing.

The implementation of Portals as a first step in a Service Oriented Architecture journey for large enterprises, however, required a level of technical maturity and business acceptance which only existed well into the 2000s. While many of the Portal implementations that existed in 2009 could have been attempted earlier, practical changes including the maturing of standards, the demands of mobile applications, the resolution of scalability issues and an understanding of usability all contributed to making the late 2000s a critical moment for Portal development.

In this book we provide an ‘on the ground’ view of some of these developments, as captured in the International Journal of Web Portals. IJWP was launched at the beginning of 2009, as a coming together of two separate approaches reflecting the professional experiences of these writers. 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.

Portals and Service Oriented Architecture in general are important areas of study given the growing complexity of modern technology systems. An innovative company can still invent a ‘must have’ customer device without reference to anyone else. Yet systems that address larger issues used by enterprises rather than individuals must interface with many other existing systems. For vendors the challenge is to relate to standards in a meaningful way while also providing compelling value in their own product. For enterprises the challenge is to protect their own existing investment in training (both of users and of technical support), infrastructure, interfaces, business systems and so on, while still being able to expect to receive the benefits of innovative technology.

This tension makes research into Portals and SOA an important area. Such research allows practitioners on both sides of the vendor-customer relationship to learn as quickly as possible what is promising and what is a dead end. While vendors themselves will provide their own view of this, vendor white
papers are unlikely to identify negative experiences with their own products, and often lack the rigor of academic research.

For our readers we have provided a balance of background information, summarising the state of a particular area, and new research that is previously unpublished. One of the pleasant surprises for us in IJWP’s first year has been the international breadth of contributions, with researchers contributing from Australia, Germany, Greece, Netherlands, New Zealand, Spain, Sweden, Taiwan and the USA.

Since launching IJWP we have established a format and set of contributors which together address these needs. In this book we have grouped the contributions by topic, rather than chronologically as follows:

- **Trends in Portal technology**: Portals are widely understood as advanced technologies. A single journal can only selectively represent technology trends. However, there are key pointers and discoveries which delineate the year’s path.

- **Service Oriented Architecture**: While a long-term approach, SOA is also undergoing its own technical evolution, one that practitioners need to understand.

- **Business issues in Portal and SOA uptake**: While addressing the general challenge of technology complexity, a key driver for Portal and Service Oriented Architecture uptake is business benefit, generally understood as cost reduction or new business opportunity. Success on these measures needs to be understood.

- **The value of case studies**: A key method adopted by IJWP to bridge the practitioner-academic gap is the use of case studies. These enable the experiences of industry practitioners, including those without previous research exposure to be reflected in a theoretically sound manner.

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

**TRENDS IN Portal TECHNOLOGY**

Portals are often considered as a window to the Enterprise. The technology and infrastructure for building portlets, either local or remote, has been evolving over more than 10 years and reached considerable maturity. We would like to review some important points of Portal technology evolution and also look at the options associated with the newly introduced trends such as Cloud technology, remote portlets and emerging gadgets.

In this technological evolution the standards and specifications (Java Specification Requests – JSR - in particular) play an important role. The first versions of Portal technology focused on basic usage with limited functionality and fast introduction to the markets, with very little consideration for Portals’ future place in integration and standards. The implementations relied on adding functionality through the maintenance process. Similarly, the standards / specifications driving the product implementation often ended up with extensions to support more advanced scenarios. At present, portlets based on JSR168 as well as JSR286 specifications are often used in Portal applications. The second version of the Java™ Portlet Specification (JSR286) brings a standard solution and interoperability to accommodate most advanced scenarios.

During the last years, we have also seen the trend to moving Portals from local web components to remotely deployed integration services (Polgar, Jana). In this book, we discuss the impact of WSRP (Web Services for Remote Portlets) 2.0 and portlet specification JSR286 for “on glass” integration paradigm.
In most cases, portlets are built to be deployed by local Portals. This is not practical if there is a demand for integration using web services in a Portal and it is expected that other business partners would 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 (Universal Description Discovery and Integration). Web Services for Remote Portlets attempt to provide solution for implementation of lightweight SOA. The UDDI extension for WSRP enables the discovery and access to user facing web services provided by business partners while eliminating the need to design local user facing portlets. Most importantly, the remote portlets can be updated by the 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 team development, upgrades, administration, low cost development and usage of shared resources.

Development of Portal software and the push for fast delivery to the markets has led to innovation in development. In the book there are several chapters authored by the researches from IBM research laboratories. The reader may appreciate the intimate insight into developing, evolving and maintaining Portal software at IBM (Thomas Stober and Uwe Hansmann). IBM’s Portal technology has been evolving for many years as a powerful infrastructure for integrating the enterprise, by presenting it as a consolidated view to all users in the company. New capabilities of WebSphere Portal 6.1 and higher are the outcome of a world-wide development team, which focused on this release. During that time major architectural enhancements have been introduced and a significant amount of code was written. The authors point out the effort of the developers and testers who have adopted Agile development principles to collaborate across the globe. The Agile approach in general is taking over in many companies. The IBM approach to developing the Portal technology using Agile approach is interesting. Developing market-leading Enterprise Portal products like WebSphere Portal requires a first-class development team. More than 300 developers and test engineers are working for different organizational units and collaborate in widely separated time zones. There are eight major development sites across the world. The product has dependencies on other IBM products, such as WebSphere Application Server, and is the base for other products, like Lotus Quickr. Further dependencies arise from customer requirements and commitments. Typically, unforeseen issues like a design flaw, a growing number of bugs beyond the expected, or redirection of resources to other activities made it necessary to rework the plan. As in the majority of large development projects, the problems are often discovered at the end, in the final test phase. The major difficulty is ensuring code stability and at the same time making significant code alterations. The delivery date cannot be postponed and sacrificing quality is not an acceptable solution.

As Portals are quickly being accepted as a focal point of an enterprise, the adaptation of Portals to Web 2.0 standards (Andreas Nauerz and, Rich Thompson) may require an innovative approach by vendors. With Web 2.0, the provision of recommendations has become an integral part of the web interface. IBM WebSphere Portal leads this wave of innovation, combining the latest user-centric functionality with reliable security and manageability features to meet the needs of business. The software incorporates extensive Web 2.0 capabilities, allowing companies to fuel social interaction by delivering high-performing, intuitive applications through a rich web interface. This new release adopts the latest industry-driven standards. It also introduces flexible ways to create and manage Portal sites and content. Many more enhancements emphasize increased utility and flexibility, such as web site management, integration of non-Portal pages as well as step up authentication. The authors propose a generic framework that allows transparent integration of different recommender engines into Portal. The framework comes with a number of preinstalled recommender engines and can be extended by adding further such components.
Recommendations are computed by each engine and then transparently merged. This ensures that neither the Portal vendor, Portal operator, or user is burdened with choosing an appropriate engine and still a high quality recommendation scan can be made. Furthermore we present means to automatically adapt the Portal system to better suit users’ needs.

Another perspective on the management of Portal development is provided by a practitioner from large companies. T. Polgar elaborates on practical approaches to the resolution of development issues and risks in a Portal environment in early stages on Portal technology. Topics discussed include implementation of Portals in an enterprise environment, portlet applications’ high availability, portlet disaster recovery, and the cost of portlet deployment. The complexity of the technology and especially the growing complexity of the development team and time constraints typically make it more and more difficult to execute the established plan as scheduled. Future needs and issues associated with enterprise business requirements are difficult to predict. Outsourcing to offshore companies makes results in development, and testing performed by separate organizations, as well as communication and interfaces between different organizational units, a challenge. It is extremely difficult to make sure that the right information is made available to the right set of people. Bringing the independently developed pieces together in order to assemble a complex use case requires a significant integration effort, before the overall system reaches a satisfactory level of stability. A typical situation can be described as a separation of responsibilities: the developers own the responsibility for design, coding and unit testing, while other project tasks are owned by different teams. As a result of these experiences, the limits of such a pre-planned waterfall approach clearly indicate that the classical approach is too inflexible to react fast enough to the highly dynamic development and testing constraints of a complex product within a large organization.

**SERVICES ORIENTED ARCHITECTURE**

Application services are characterised by richer functionality and content than normal specialized services. They scale up the concepts of service-oriented computing to business-level so that service-oriented computing concepts became more receptive to business needs. Technically, application services can be composed and configured to service different but similar domains. An application service hides the implementation detail from its users and it can be a composition of other services. To construct an application service is typically a labor-intensive and error-prone process. Moreover, application services developed through traditional development methods expose the same problems associated with late delivery, over budget, unpredictable quality, lack of reuse and so on. Clouds are a next generation of infrastructure that is based on virtualization technologies such as virtual machines. Clouds are able to dynamically provision services on demand as a personalized resource collection to meet a specific service requirement, which is established through negotiation and which is accessible as a service via a network. Paul Cooper and Matthew Hodgson introduce the perspective of a large consulting company which is typically involved in many integration projects with tier 1 enterprises.

In this book, we have also looked at what other people expect from SOA and Enterprise Service Bus (ESB). Our findings are summarized as follows: Many early SOA-based implementations have been built on EAI, J2EE- and .NET-based middleware, including message brokers, application servers, and enterprise service buses. Increasingly, however, data integration has become a primary objective. There is a growing awareness that a data integration platform should enrich a SOA solution with sophisticated data services beyond the scope of application integration-centric technologies. Over the past years, data
integration technology has evolved with built-in support for XML transformations, Web services protocols, JDBC connectivity, and Java Message Service (JMS) connectivity. Advanced data integration platforms also feature metadata capabilities driving the core of their development and run-time infrastructure. The metadata provide an abstraction of the business logic from the technical implementation, and enable delivery of advanced data integration functionality over the data services layer to many components. The integration platform functionality can be packaged and reused across multiple projects to reduce development and deployment costs. Integration logic in the ESB should eliminate the need to hand-code data integration connectivity, and enable businesses to realize rapid time to market.

Delivering Portals and gadgets to the Enterprise quickly is enabled by the introduction of Web 2.0 standards, particularly gadgets which are widely loved. They are everywhere, on desktops, mobiles and elsewhere. Gadgets enable several social activities: to keep track of a social network, finances, work duties. They are just part of everyone’s life. How can we bring this technology to your enterprise? For example, dashboards can be easily assembled and customized to keep track of business activities. This is the technology of the future, and Portal technology is providing the basic components on which the next gadgets can easily be built. Mobilizing the enterprise (Andrew E. Young) is a chapter which examines current mobile Service Oriented Architecture (SOA) research concerns and presents approaches to the challenges of enterprise support for mobility.

Leveraging web services through Portals by means of the Java Portlet and WSRP standards gives companies a relatively easy way to begin implementing SOA. Today, most Portal servers as well as development environments 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 architects to easily create and offer SOA-style services solutions. These services are then published in order to provide easy access and adoption by other service 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 other consumers in a fashion which enables the consumer to conveniently consume the services and use them without any programming effort. Furthermore, an Enterprise Service Bus can be used to create a controlled messaging environment, thus enabling lightweight connectivity. However, architects should be aware of some issues associated with use of current web technologies: cookies handling, cookies protocol, URL rewriting rules, Ajax, and security handling. In this book, we provide an in-depth overview of the advantages and disadvantages in deploying these technologies in conjunction with WSRP.

The potential benefits and current problems of web services are often discussed in academic articles and, less commonly, in books. Quite simply, the vision projects that software functionality can be made available over the Internet and consumed as a service by clients regardless of their architecture, language, or communication protocol. Standards have been agreed to enable this vision to be realized, principally by using UDDI for publishing and discovering services, SOAP for communication, and WSDL as the description language. OWL-S is emerging as the standard for capturing the semantics of service operations and BPEL4WS for composing atomic services into workflows. A popular approach to solving the terminology problem is to express service details using the OWL-S framework for web service descriptions and ontological reasoning techniques derived from AI to match requests with advertisements. This allows semantically equivalent terms to be treated as such, despite syntactic differences, and matching to be a matter of degree rather than an all-or-nothing affair. Many more accounts of these standards exist (Polgar, Willkinson).
A contemporary architectural approach for an orchestrated, agnostic, federated enterprise through the adoption of loosely-coupled open Service interfaces is presented and critically evaluated. The Service Oriented Architecture (SOA) paradigm unifies disparate, heterogeneous technologies in an attempt to resurrect legacy technology silos with a Service ‘face-lift’ (Andrew Young). In this book we explore current views, and critically review a variety of research papers in this field. Two chapters included in this book draw extensively on published research in the past two years. The second of these concentrates on the technology of SOA particularly, Semantics, Representational Start Transfer (REST), Object Orientation and Operations and Quality aspects. These views are introduced by discussion and explanation of REST technology by Jan Newmarch. As Dr. Newmarch explains, the REST approach was developed to describe the architecture of distributed resource access, such as the WEB. REST in comparison with WS-*stack works with the web rather than against it and is getting increasing support, not only from developers but also from vendors. This paper explains the philosophy of REST and highlights its simplicity in accessing resources.

In the first issue of the *IJWP*, we introduced the perspective on strategy for enterprise-wide alarming. Andrew Young explains that the demand for contemporary IT systems to support availability, expansive integration and extensibility requires strategic and well-thought approach to monitoring the servers. Distributed infrastructures and particularly the advent of Service Oriented Architecture (SOA) introduce new challenges for meeting these demands. Despite architectural conventions to prescribe a common structure and simplify approach, these systems are becoming more complex, heterogeneous and critical. Comprehensive System Management is no longer a luxury. Faults and potential failures have to be identified, isolated and addressed, ideally pre-emptively. In every enterprise, the front-line indicators are alarms. Single points of failure need to be identified and eradicated through redundancy and balanced resource allocation. Despite these efforts, there will always be potential for a system to fail. An alarming strategy serves to identify potential weaknesses and mitigate their consequences. The prevalence of SOA, and distributed architectures in general, utilising loose coupling and dislocated services while promising little or no disruption to service makes failure analysis and mitigation more important than ever. Although it is not possible to develop a comprehensive alarming strategy without a detailed knowledge of the system to be monitored the author elaborates on currently available alarming Instruments and approaches and evaluates associated strategies and implementations.

**BUSINESS ISSUES IN PORTAL AND SOA UPTAKE**

Portals have the potential to provide every user with a unique and even user-customisable experience. While a small intra-organisation user-created Portal may require little additional effort and cost, a fully functional, secure, highly usable public Portal will cost millions, if not tens of millions of dollars to design, test and implement. Service Oriented Architecture is a step beyond this. A major enterprise seeking to service enable their key functions and create an integrated environment will be looking at tens or hundreds of millions of dollars over several years.

Despite these costs the promise is significant. Increased customer self services translates into reduced customer support costs. User-customisable environments are sticky: the customer will be more likely to spend more time in an environment they have already committed time to create. On the SOA side, parameterisation has been a long held dream: the most simple change to core system code can take millions of dollars and a year to implement, given testing and scheduling requirements. If a company
or government department could ‘plug and play’, take existing functionality and assemble it into a new service in weeks rather than years, this could make a huge difference to a company’s market position or a government’s response to voters’ interests.

Most of the contributions to *IJWP* provide details of the benefits of a particular technology or approach, and these can be taken as suggestions of business benefits. Separate from these two chapters (Adamson; Clohesy, Frye & Redpath) have focussed primarily on understanding where Portals and other SOA-based applications sit in a longer term business context.

Adamson compares the business opportunities of Portals and Service Oriented Architecture to two previous periods of business focus on technology. The 1980s technology Electronic Data Interchange (EDI) provided a set of standards and approaches that for the first time allowed the exchange of business data between commercial partners or even anonymously based on standardised message formats. EDI proved highly effective for large enterprises, but very poor (due to cost and complexity) for small organisations in most industry sectors. The rise of the Internet’s commercial potential in the mid-1990s was followed by the significant promise of ‘friction free capitalism’ through the introduction of Business to Business (B2B) trading exchanges. These universally failed to meet expectations, although a very small number reinvented themselves to provide more modest services. While the Internet’s ubiquity laid the basis for B2B, their business models failed to meet the requirements of actually existing commerce. The enterprise Portal takes up the standards-based sectional success of EDI and widespread connectivity that underpinned B2B exchanges. It is too early to say whether that promise will be met. However, the experience of EDI and B2B provide us with guides in measuring Portal success.

Clohesy, Frye and Redpath focus on applying a more systematic approach to business architecture. They argue that while technical architecture is well developed and understood, when it comes to applying this in a business context, there is no equivalent systematic approach to the development of business services. The writers suggest that based on the experience of technical interoperability we should be able to similarly approach a business’s processes as structured assets that require formalisation. This is done through a proposed approach of the conceptual business service.

Additional context is provided through a thorough review of Portal history (Tatnall). Alongside commercial enterprise Portals, the not-for-profit sector is a major user of Portals, particularly the educational sector. Tatnall describes the application of Portals in a university environment (Tatnall), and Manouselis et al. give a detailed discussion of one educational Portal, Organic.Edunet.

One further business challenge that we saw through 2009 was a continued tightening of enterprise investment in new technology return on investment began to be understood as achieving payback in three or at most six months. Organisations to some extent lost interest in maximising the benefit of their technology investments as they focussed on their own short and medium term survival. This has had the unfortunate consequence of minimising experimentation, with an inevitable impact on new areas of investment. While this is not a universal phenomenon, it has meant that some expected developments have slowed or stopped, and even organisations in a strong position are willing to take a wait and see approach.

**HOW CASE STUDIES HELP OUR UNDERSTANDING**

For Stake (1995) case study research focuses on the particular, and this can then provide further understanding of an issue or problem. We published three case studies in 2009. These were found to be use-
ful in bringing practitioner experience to the reader relatively rapidly. Two of these studies (Brewer & Adamson, Worley & Adamson) included here took the form of a set of transcribed open-ended questions between one of the writers and industry practitioners. These practitioners deal with the ‘sharp end’ of Portal and SOA implementation. Going beyond the expected experience, these practitioners described the difficulties and opportunities that emerged in practice as they sought to implement technology solutions in a business environment.

Daniel Brewer focuses on understanding the reasons for Portal implementation, and then looking at challenges that arise, especially in the support and security areas. Both of these emerge because of the flexibility provided to end users. If an end user is allowed to transform the look of their workspace, as Portal technology allows, then when they have a problem they call a help desk, which, in the worst case, may have little understanding of the environment a user is working in. Security poses a similar challenge, as a more complex environment increases the challenges of managing user access and other security requirements. Brewer also speaks of the approach to Portals provided by Microsoft SharePoint. At the time of writing Microsoft had not attempted to achieve the standards based rigor of other enterprise Portal products. Nevertheless, or perhaps because of this, it had gained significant market share through its ease of implementation and pricing model.

Brenton Worley describes his experience in relation to Service Oriented Architecture implementations in the enterprise space, one in the utility sector and the other in retail. In each case the business driver for SOA adoption was a changing business environment. For the utility sector this is driven by external regulatory forces, while in the retail example the client hoped to maintain an industry lead while extensively upgrading their IT infrastructure. What Worley had found in both cases was a mismatch between tools and needs. The SOA tools provided by vendors appeared to address problems that were years ahead of the actual customer challenges. While customers were seeking the best level of granularity for service definition, vendors were offering automated solutions for complex SOA-based environments. This in turn led to a mismatch: the picture painted by vendor promise, and understood by customer senior management as immediately achievable, described a world far beyond the current actual customer capability.

A third case study was provided by Tatnall and Burgess. This provides a longitudinal view of two distinct Portals, one in Australia and the other in Bangladesh. The initial data had been collected in the early 2000s, and followed up in 2009. Both cases deal with business to business Portals, a horizontal B2B industry Portal in Melbourne, and a vertically integrated industry Portal in Dhaka. Such a study is able to examine question of business benefit, as well as the phenomenon of shifting technology use, where usage itself changes the purpose of technology over time. A summary of ‘theory versus reality’ in the study lists six aspects and details the outcomes for each. This provides valuable insights into the changing expectations for B2B Portals between 2000 and 2009.

In summary, the case study method has been useful in early identification of practical challenges in the implementation of Portals and Service Oriented Architecture. Given the cost of implementation of even modest sized enterprises, the impact on organisations when changing underlying architecture, and the rapid changes in consumer technologies over the same period, timely identification of benefits and challenges is highly valuable in this field. Findings that can be fed back into the planning process for Portal and SOA implementations have the potential to significantly reduce unnecessary costs.
CONCLUSION

Today the Web is used as a means to allow people and business to use services, get information and conduct transactions. Businesses today depend upon their visibility in their respective marketplaces and provision of e-services to customers. The Internet has become an important delivery mechanism of business visibility. The Internet also significantly extends business’ capabilities to sell and buy worldwide. The company website plays an important role in maintaining and extending the business opportunities over the Internet. Furthermore, application services represent multi-billion dollars of markets in the IT industry. Both application service providers and consumers invest a considerable amount of time and effort to develop application services-based solutions. Many package based business applications such CRM or financial services are critical for business success and their fast deployment and customization offers significant business benefits. This demand places an ever increasing pressure on leading package application providers such as HP, IBM and SAP to provide domain-specific skills, and develop a packaged solution for any type of the business in order to help their clients reduce cost and mitigate risk. However, this approach is encountering its own limitations. First, project planning and implementation for application services are time-consuming and costly. It requires a variety of skills and expertise that many companies do not possess. There is also a high cost associated with the ongoing management and maintenance of these package based applications. Packaged applications are often tightly integrated with existing systems, and clients typically require a broad range of technical expertise to run them, which is usually hard or expensive to obtain from vendors or consulting companies. With major concerns about the cost, both service providers and consumers experience the difficulties characterised by the attempt for more and more cost reduction. Consequently, it leads to reduced service quality rather than to reduced cost of the solution. There is no room left for improvement, and typically, with the improvement of packages and new versions, the customization efforts eat more money.

All this leads to an ever present requirement for application services to provide richer functionality and content than a package can deliver with its specialized services. We often talk about Service Oriented Architectures (SOA) as a magical integrator capable of delivering the application services as a seamlessly integrated component. An application service hides the implementation detail from its users and, by itself, can be also a composite of other services. To construct an application service is a labor-intensive and error-prone process if the package is used or the service is developed using company internal domain expertise. Moreover, integration of many application services developed through traditional development methods exposes the same problems associated with most development processes in enterprise applications, late delivery, over budget costs, unpredictable quality, lack of reuse and so on.

Portals and SOA based applications promise to resolve these issues. This is accompanied by other innovation including Cloud computing, technology based on virtualization technologies such as virtual machines supporting dynamic provisioning of services on demand and personalization of services. Will these challenges and promises be met? The answer depends on understanding both the technical detail and the vendor and user experiences. Through the contributions in this book we hope to assist in this understanding.

Jana Polgar
NextDigital, Australia

Greg Adamson
University of Melbourne, Australia