Jornada de Seguimiento de Proyectos, 2004
Programa Nacional de Tecnologías Informáticas

CONCEPCIÓN Y SOPORTE DE PORTALES PARA PYMES DE FABRICACIÓN
TIC2002-01442

Oscar Díaz García

Abstract

A portal is a key component of an enterprise integration strategy. It provides integration at the user interface level, whereas other integration technologies support business process, functional or data integration. To this end, *portlet syndication* is the next wave following the successful use of content syndication in current portals. A portlet is a full-fledge application which is rendered within the portal framework. Looking at portlets as the main portal components, this project looks into three main issues: portlet adaptation, portlet-izing web application and portlet orchestration.
Keywords: Portlets, portals, components, web application development.

1 Project aims

A portal is a key component of an enterprise integration strategy. Key to this view is the notion of *Portlet. A Portlet is a multi-step, user-facing application to be delivered through a Web application.* Portals as application-integration platforms have rapidly embraced this technology, and they are currently the most notable Portlet consumers[1].

So far however, the lack of a common model prevented Portlet interoperability. This impedes a Portlet developed in, lets say, *Oracle Portal,* from being deployed at a *Plumtree* portal, and vice versa. However, the recent delivery (2003) of the *Web Services for Remote Portlets* (WSRP) specification [?] promises to overcome this problem.

WSRP uses WSDL for Portlet specification. Unlike traditional Web services, Portlet operations might not only return raw data but fully rendered markup (e.g. XHTML). This markup is then included within a portal page, with very few changes to be made by the portal. For the purpose of this paper, this is referred to as *"Portlet Syndication"* in comparison with the well-established term, *"content syndication".*

Content syndication is currently a common practice to enrich the offerings of a portal. Yahoo is a case in point. Most of its features come from other Web sites: travel, weather, maps or searching services based on Google. "All" Yahoo does is nicely aggregate these outputs together. A similar approach can now be followed for full-fledged applications: the Portlets. As an example, consider Iberia, the Spanish airline. It offers the *ticketPurchase* application as a WSRP Portlet that encapsulates both the functionality (e.g. access to *Iberia's* database)

[1] Portlets are endorsed and supported by IBM, BEA, Oracle, Sybase, Viador, Verify, PeopleSoft, Plumtree and Vignette. Portlets are also referred by some vendors as *Web parts* or *gadgets.*

and most important, the presentation and control-flow strategy that characterise the process of ticket purchase. This Portlet can now be readily syndicated into WSRP-compliant portals (e.g. a portal of a travel agency) without this portal being concerned in all the intricacies involved in how to display the process that leads to a ticket purchase.

Regarding portlets as main portal components, this project looks into three main issues, namely, portlet adaptation, portlet-izing web application and portlet orchestration. The former is shared with other component technologies. To improve the reusability of portlets like any other COST, portlets should be engineered for adaptation. The second issues is concerned from accessing web applications from a portal. Over the last few years, organisations have invested in web-enabling some of their existing systems, and they can be interested in capitalising on these Web applications by wrapping them as portlets so that they can be plugged into third-party portals. Finally, portlet orchestration has to do with seamlessly integrating the distinct portlets which are available within a portal. The challenges pose by each of these issues are addressed throughout this report.

So far (i.e. 18 months since the beginning of the project), the time dedicated to accomplish each of these goals follows:

- portlet adaptation: 18 months
- portlet-izing web applications: 12 months
- portlet orchestration: 12 months

During the rest of the project, we plan to re-inforce the second aim (i.e. potlet-izing web applications) due to its potential to be transfered to industry.

2 Project accomplishments

This section is arranged along the three main aims of the project i.e. portlet adaptation, portlet-izing web application and portlet orchestration. Each subsection includes a brief description of the problems encountered, and how these difficulties have been addressed.

2.1 Portlet adaptation

Portlet syndication raises variability concerns. The notion of variability indicates the extent to which a Portlet can be seamlessly integrated into distinct utilization contexts, e.g. distinct portals. For example, the *ticketPurchase* Portlet can be syndicated to distinct travel agencies. The processing context is the same, i.e. *Iberia's* infrastructure, but the utilization context varies for each agency. An agency located in Spain will use Euros as the standard currency, impose a red background on the *ticketPurchase* markups, and restrict the flights to domestic ones. By contrast, an agency in New York will use dollars, impose a gold background, and focus on international flights.

The premise of this work is that the diversity of the settings where a Portlet might be syndicated, recommends Portlets to be instrumented for variability. That is, effective Portlet syndication demands an analysis of the commonality and variability of the distinct scenarios to be supported, and how this variability is going to be upheld in a cost-effective manner.

The stringent demands on Portlet variability are not new in the component realm. Over years of evolution of a software system and also during customisation for reuse, components are affected by changes stemming from new requirements, new versions of operating systems, etc. If it is necessary to maintain a component version for each combination of those variants, components will grow in size and number. The cumulative effect of this uncontrolled growth may make it prohibitive to reuse components. *"This situation is particularly severe for application-level business components that must accommodate a wide range of variant requirements reflecting different reuse contexts"* [?]. This is the case for Portlets.

2.1.1 Our contribution

To overcome this situation, the paper introduces a product-line approach to Portlet development. Product-line techniques have been successfully applied to a broad range of domains [?]. However, to the best of our knowledge, this is the first attempt to apply this technique to Portlet development.

A product-line approach begins by assessing the variability requirements the product line needs to support. This can include contextual features such as the locales, the user agent, etc. This paper introduces a new source of variation (i.e. a feature) more akin with the Portlet idea: *the interaction lifecycle.*

The interaction lifecycle captures variations on the *visible* process flow of the Portlet (e.g. using GUI terminology, the *breadcrumbs* that lead to the purchase of the flight ticket). Hence, the idiosyncrasies of the potential Portlet Consumers are captured in terms of variations on the interaction lifecycle of the Portlet.

A source of variation (e.g. user agent) has variants (e.g. *Netscape702*, *msie60*, *nokia7650).* A distinctive characteristic of this approach is that the variants of the *interaction lifecycle* are categorised based on the freedom left to the Consumer to customise the final product to its own idiosyncrasies. The Portlet Producer facilitates "the hooks", and it is left to the Consumer to decide what *amendment* "to hang" from these hooks to produce a custom markup. This work introduces distinct types of amendments which permits the Consumer to customise the style, the content or the control-flow of the markups that realise the *interaction lifecycle.*

This approach permits the separation of the processing context (Producer) from the utilization context (the Consumer) which is a decisive mechanism for managing partnerships. An approach which does not have this property would lead to an impasse, as it would be impossible to simultaneously deploy the processing in heterogeneous utilization contexts at the partners' companies [?].

2.2 Portlet-izing web application

Over the last few years, organisations have invested in web-enabling some of their existing systems. Also, the Web itself is now a rich source of information and applications. Both aspects vindicate the effort of being able to tap into these resources, making them available from the portal.

Invoking a Web application from a portal poses no problem. An *<href>* tag can do the job. However, *integrating* a Web application into a portal is a different matter. Integration entails a stronger weaving with the functionality offered from the portal framework so that the End-user enjoys the commodities offered by the portal when working with the Web application.

This integration can be shallow or deep. A shallow integration makes the "home page" of the Web application accessible from the Portal. This is also known as "web clipping", and tools are currently available in main Portal IDE providers such as Oracle Portal or IBM WebSphere [?, ?].

Web clipping allows to obtain a doorway to the Web application from the portal. As an example, consider a Web application such as *www.easyjet.com*. A shallow integration implies to be able to display the home page of *easyjet* from the portal rather than rendering this page in a separated browser agent. This permits the profiles stored at the portal framework to be applied to the *easyjet* home page (e.g. a default value for the *departure airport* parameter), and in so doing, improves the user experience. However, once this doorway is crossed, the customer moves to the Web application, and a new browser agent is open. Now, you are no longer within the Portal realm.

By contrast, a deep integration makes the entire application -not just the home page- available from the portal framework. If the *easyjet* site is deeply integrated into the portal framework then, all interactions with *easyjet* are achieved without leaving the portal. Even more, the *easyjet* functionality should be supplemented with the customisation or identification commodities offered by the portal framework. From an implementation viewpoint, this implies to wrap the Web application as a portlet.

2.2.1 Our contribution

Besides all the technicalities that go to make an *external Web application* behave as a Portlet, our work focuses on **customisation issues**. To this end, the term **bridge Portlet** is coined to refer to a Portlet that permits to wrap a Web application as a Portlet. That is, the Portlet, which follows the bridge pattern [?], handles the Application as a black box, and accesses it only using URL requests, while the Application treats the Portlet as any other Web client.

Customisation has been defined as *"the adaptation of an application's services towards its context"* [?]. Kappel et al. characterise adaption in terms of the kind of adaptation (i.e. what changes have to be done), the subject of adaptation (i.e. what to change), and the process of adaptation (i.e., how adaptation is performed). Next paragraphs address these questions for the current work.

The kind of adaptation. Broadly speaking, a Web application is an aggregate of *pages*, whereas a Portlet is an aggregate of *fragments*. A main difference is that a fragment should normally share the page space with other Portlets. Hence the content of a page can not be framed within the region available at the portal end. The mediator can resort to filter out some of the content/images of the Application output which are not strictly necessary to proceed with the Application flow. For the *easyjet* example, this implies that the Portlet markup should focus on the elements that pursue the goal of buying a ticket, while removing banners, navigation bars, etc. Another alternative is for the returned page to be fragmented and gradually presented. In this case, a Portlet interaction (i.e. an End-user click on the Portlet fragment) does not necessarily lead to a new *http* request to the remote application. This option implies the Portlet to cache the HTML page, split it into different fragments, and render a fragment at a time.

Besides removing some markup, the content of the Application might need to be supplemented with additional markup that support the extra functionality required for the Applica-

tion when syndicated within the portal. Bookmarking[2] is a case in point.

The subject of adaptation. This is characterised by looking at the level of the web application which is effected by the adaptation (i.e. content, navigation or presentation) as well as at the concrete elements (e.g. pages, links, input fields, etc).

Turning pages into fragments can affect all the content (e.g. addition/removal of markup chunks), navigation (e.g. a page is split into distinct fragments) and presentation (e.g. CSS mapping with the portal guidelines is needed).

The process of adaptation. This work strives to support automatic adaptation (i.e. no user involvement), and deferred dynamic adaptation (i.e. the adaptation is done not before the user requests the page which is subject to adaptation).

2.3 Portlet orchestration

Aggregating Portlets into a portal is more than merely invoking these services, or arranging their fragments together in the same portal page (i.e. the so-called *"side-by-side"* aggregation). Information contained in one Portlet will surely be required in another, and requiring the individual user to manually copy and key in data from source to target Portlets leads to frustration, lost productivity, and inevitable mistakes. And this situation certainly hinders the fulfillment of the portal imperative.

To address this limitation, distinct mechanisms have been proposed which can be classified as data-based and API-based. The former permits distinct Portlets share a common piece of information but within the scope of the same provider. Portlets which pertain to distinct providers remain isolated. On the other hand, the API-based approach facilitates a programmatic interface for Portlets to communicate their state to interested parties. Unfortunately, there is not yet an agreement on how to standardize this mechanism. Indeed, standardizing this API would lead to commoditize one of the most valuable offerings of portal vendors. Hence, vendors might be inclined to retain this competitive advantage rather than commoditizing it and enabling other companies to exploit their application logic and infrastructure functions on top of it.

2.3.1 Our contribution

This project address an **ontology-based** approach to Portlet aggregation can overcome some of these drawbacks. The proposal is characterised by (1) making explicit the Portlet ontology, and (2) deep-annotating Portlet fragments. This mechanism makes explicit what is hidden in the data-based approach, and unlike the API-based proposal, requires no agreement with other Portlet providers.

The key aspects of the approach can be summarized as follows:

1. Portlet applications are characterised by their ontologies. Although none of the Portlet standards (i.e. WSRP [**?**] and JSR168 [**?**]) contemplate this option, the extensibility mechanisms available in both standards can be used to extend the Portlet description with an additional *ontology* property. Besides facilitating Portlet integration, all the benefits

[2] By bookmarking in a Portlet context, we mean generating a bookmark for a previously visited Portlet fragment, independently of the portal context. Further reading in the intricacies of bookmarking can be found at [**?**].

of using explicit ontologies (e.g. better documentation, search, knowledge acquisition) are brought to the Portlet realm.

2. Portlet fragments extend their markups to permit interested parties to gain access to the *delivered data* and *prompting actions* that the fragment supports.

3. Portlet aggregation is achieved through deep annotation. Deep annotation has been proposed in [?] as an annotation process that *"utilizes information proper, information structures and information context in order to derive mappings between information structures"*. Here, this mapping serves to indicate how content from one Portlet "flows" to a neighbouring Portlet.

3 Project-acomplishment metrics

3.1 Publication accomplishments

Journals

1. O. Díaz, J.J. Rodríguez (2005) *Portlet syndication: raising variability concerns*, ACM Transactions On Internet Technology, Vol 5, Nž 5, August, 2005

2. O. Díaz, J.J. Rodríguez (2004) *Portlets as Web Components: An Introduction*, Journal of Universal Computing Science, Vol 10, Nž 4

3. O.Díaz, F. I. Anfurrutia (2004) *On Derived Elements in XML documents*, Journal of Electronic Commerce and Research Applications, Accepted for publication.

Conferences (acceptance rate below 30%)

1. O. Díaz, I. Paz (2004) *Invoking Web applications from portals: customization implications*, Int. Confer. on Adaptive Hypermedia, 3137 LNCS, Springer-Verlag

2. S. Trujillo, I. Paz, O. Díaz (2004) *Adaptability in Portlet Frameworks: a comparison* (short paper) Int. Conf on Web Engineering (ICWE), LNCS, Springer-Verlag

3. O. Díaz, C. Calero, M. Piattini, A. Irastorza (2004) *A portlet usability model*, 2nd International Conference on Service Oriented Computing (ICSOC), IBM technical report

4. O. Díaz, I. Paz (2004), *Portlet-izing Web Applications: raising the issues along the WSRP Standard*, 2005 International Symposium on Applications and the Internet (SAINT), IEEE Computer Society Press

5. Jon Iturrioz Oscar Díaz, Sergio F. Anzuola (2004), *Facing Document-Provider Heterogeneity in Knowledge Portals*, 6th International Conference on Advanced Information Systems Engineering, (CAiSE 2004), 3084 LNCS

6. O. Díaz, S. Trujillo (2003) *Moving co-branding to the Web: service-level agreement implications* 4th. IFIP Working Conference on Virtual Enterprise (PRO-VE),Kluwer Publishers

7. O.Díaz, S. Trujillo, I. Azpeitia (2003) *User-FacingWeb Service Development: a case for a product-line approach*, 4th VLDB Workshop on Technologies for E-Services (TES'03) 2819 LNCS

8. Sibylle Steinau, Oscar Díaz, Juan J. Rodríguez, Felipe I. Anfurrutia (2003), *A tool for assesing the consistency of Websites*, Enterprise Information Systems IV, Selected papers form ICEIS'03. Mario Piattini and Joaquim Filipe and Jos{\'e} Braz (Ed.) Kluwer Academic Publishers, ISBN 1-4020-1086-9

Conferences (acceptance rate above 30%)

1. Jon Iturrioz, Oscar Díaz, Sergio F. Anzuola, Iker Azpeitia (2003), *The Semantic Desktop: an architecture to leverage document processing with metadata*, Proceedings of the VLDB Workshop on Multimedia and Data Document Engineering (MDDE'03),

National publications

1. Juan J. Rodríguez Oscar Díaz, Iñaki Paz, Felipe I. Anfurrutia, *Web Application Syndication en Oracle Portal*, XIII Congreso nacional de usuarios de Oracle, 2003

2. Jon Iturrioz, Oscar Díaz, Sergio F. Anzuola, *Coupling the ontology layer with the resource layer:a rule-based approach*, VIII Jornadas de Ingeniería del Software y Bases de Datos, 2003

3. Oscar Díaz, Juan J. Rodríguez, *Applying product-line techniques to enhance portlet adaptability.* Taller de las JISBD sobre Desarrollo y Mantenimiento ágil de aplicaciones basadas en Servicios Web (ZOCO), 2004

Submitted to publication

1. Oscar Díaz, Juan J. Rodríguez, *Portal Application Development: a Service-Oriented Approach*, Submited to ACM Transactions on Information Systems, 2003 (second review)

3.2 Training accomplishments

The team currently includes 4 PhD students and 2 technical assistants. Last year, one of our students got an "european" PhD, specificaly, J.J. Rodríguez, read his PhD. entitled *Portal Development Using Interactive Web Services.* The tribunal was formed by Prof. A. Buchman (Darmstadt, Germany), Prof. K. Aberer (Lausanne, Switcherland), Prof. A. Olive (Barcelona, Spain), Prof. M. Piattini (Ciudad Real, Spain) and Prof. O. Pastor (Valencia, Spain). As externals, Prof. N. Paton (Manchester, U.K.) and Prof. M. Bouzeghoub (Paris, France).

3.3 Technology-transfer efforts

We are proud to acknoledge that J.J.Rodriguez is currently the R&D leader at LKS Consulting, a main firm in the Basque Country. LKS recognises the abilities and expertisse of Dr. Rodriguez in portal design and development which were seen as very valuable for one of the business area of the company.

LKS is currently supporting the activities of the group through 50.000 euros.

3.4 Colaboration with other groups

We outstand the following visits to San Sebastián:

- Dr. M. Piattini and Dr. C. Calero from University of Castilla-La Mancha. Aim: apply usability measures to portlets. Results: a join paper presented at ICSOC, and a 6-month stage of a PhD. student of Castilla-La Mancha in San Sebastián
- Dr. A. Vallecillo from University of Malaga. Aim: appy component-based techniques to portlets. Results: so far there is not any visible result
- F. Bellas from University of La Coruña. Aim: portlet development using the eXo platform. Results: so far there is not any visible result