Journal of Librarianship and Information Science

<http://lis.sagepub.com>

Evaluating open source portals

Dion Goh, Brendan Luyt, Alton Chua, See-Yong Yee, Kia-Eng Poh and How-Yeu Ng
Journal of Librarianship and Information Science 2008; 40; 81
DOI: 10.1177/0961000608089344

The online version of this article can be found at:
<http://lis.sagepub.com/cgi/content/abstract/40/2/81>

Published by:

<http://www.sagepublications.com>

Additional services and information for *Journal of Librarianship and Information Science* can be found at:

Email Alerts: <http://lis.sagepub.com/cgi/alerts>

Subscriptions: <http://lis.sagepub.com/subscriptions>

Reprints: <http://www.sagepub.com/journalsReprints.nav>

Permissions: <http://www.sagepub.co.uk/journalsPermissions.nav>

Citations <http://lis.sagepub.com/cgi/content/refs/40/2/81>

Dion Goh is an Associate Professor at the School of Communication & Information, Nanyang Technological University. He is also the Director of the MSc in Information Science programme. Dion's research interests are in portal and digital library applications, information retrieval, Web and text mining, and the use of information technology in education.

Brendan Luyt is an Assistant Professor at the School of Communication & Information, Nanyang Technological University. His research interests focus broadly on the social and policy landscape of information access. He teaches in the Information Studies programme at NTU.

Alton Chua is Assistant Professor with Nanyang Technological University (NTU) where he teaches in the MSc in Knowledge Management and MSc in Information Systems programmes. Alton's research interests lie mainly in communities of practice, knowledge management strategy, leadership and online education.

See-Yong Yee, Kia-NGoh Poh and How-Yeu Ng are graduates of the MSc programme in Information Systems at Nanyang Technological University, Singapore.

Evaluating open source portals

DION GOH*, BRENDAN LUYT, ALTON CHUA, SEE-YONG YEE,
KIA-NGOH POH AND HOW-YEU NG

Portals have become indispensable for organizations of all types trying to establish themselves on the Web. Unfortunately, there have only been a few evaluative studies of portal software and even fewer of open source portal software. This study aims to add to the available literature in this important area by proposing and testing a checklist for open source portal software evaluation. The study describes the experience of evaluating open source portal software for developing a portal for cultural information exchange. Using the checklist, four candidate open source portal software packages, namely Jetspeed, uPortal, Liferay Portal and eXo Platform were evaluated. Results indicated that uPortal was the best performer followed by eXo platform, Liferay and Jetspeed. The present work represents a first step in the development of an exhaustive evaluation tool for portals. While the focus of this paper is entirely on open source software, the checklist can also be readily applied to commercial products.

KEYWORDS: checklists; evaluation; open source portals

INTRODUCTION

Commonly defined as single-point-access software systems which are intended to provide easy and timely access to information and to support communities of users who share common goals (Mack et al., 2001), portals are fast becoming an indispensable platform for any modern-day organization that seeks to leverage the ubiquity of the Web. Unfortunately, there have only been a few evaluative studies of portal software and even fewer of open source portal software. This study aims to add to the available literature in this important area by proposing and testing a checklist for portal software evaluation.

The study builds on the experience of one particular portal design project, namely, the Asia-Europe Foundation (ASEF) portal. Headquartered in Singapore, the ASEF was launched in 1997 with the aim of building better understanding between the peoples of Asia and Europe through intellectual and cultural activities, including exchanges and public events. In December of 2003, at the foundation's Conference on Culture and Civilizations in Beijing, it was suggested that a portal for cultural information would be a useful project for the foundation to undertake. Three research reports were subsequently prepared. One of these reports deal with the cultural resources of Asia (ASEF, 2005) and another with the cultural resources of Europe (Fisher and Varbanova, 2005). The third report concerned itself with the technological aspects of the portal (Goh, 2005). This third report recommended the use of open source portal software to fulfil the aims of ASEF.

Open source software allows for free distribution and redistribution of software and source code, and is developed by a community of volunteers,

* To whom all correspondence should be addressed.

Address: Division of Information Studies
Wee Kim Wee School of Communication
& Information
Nanyang Technological University
31 Nanyang Link
Singapore 637718
Tel: +65 6790 6290
Fax: +65 6790 5214
E-mail: ashlgoh@ntu.edu.sg

Copyright © 2008 Sage Publications
(Los Angeles, London, New Delhi and Singapore)
Vol 39 (3): 81-92
[DOI: 10.1177/0961000608089344]

including software developers and experts from various fields with different bodies of knowledge. There are several advantages associated with open source software such as rapid progress because of the large numbers of the contributors and reliability of the software due to the peer review given by like-minded professionals with no commercial agenda (Von Hippel and Von Krogh, 2003). Moreover, the portal could easily be used as a platform to test new concepts and technologies given the availability of the source code. For these reasons, it was decided in the early stages of the project that open source portal software was to be used.

As part of the preparation for this third report, a review of the literature of portal evaluation was conducted. Although some studies, mainly conducted by institutions and software vendors, attempted to tackle the problem of portal evaluation, none focused specifically on open source portal packages. It was therefore decided that a method for evaluation would be developed and used to test a number of candidate open source portals. In particular, the research focused on the following:

- 1) Developing a checklist of functions and features that mark a good open source portal package. This was done by incorporating not only the features identified in the third ASEF research report, but also those listed in the literature evaluating commercial portals, for example Butters (2003) and Keller (2001).
- 2) Evaluating the functions and features of four candidate open source portal tools to be considered for the ASEF project: Liferay, UPortal, Jetspeed and eXo Platform. Evaluation will be based on the checklist in order to recommend, if possible, one particular package as the best of its class.
- 3) Identifying and comparing the strengths and weaknesses of the selected four portal packages given the outcome of the checklist evaluation.

This paper first presents an overview of open source software and the available literature of the evaluation of portals. Thereafter, it describes the method used in the study and discusses the findings. In conclusion, it outlines the limitations of the study and suggests further avenues for future research.

OPEN SOURCE SOFTWARE

Open source portals are quickly gaining support and recognition as an alternative to commercial portal software. The main reason is that commercial portal software has a high total cost of ownership and provides little opportunity to explore and truly test the software prior to acquisition. Commercial software often comes with a hefty purchase and support price, and there is always the

risk that vendors may discontinue the product or go out of business.

In addition, open source allows for greater freedom in software usage and development because any programmer with the requisite skills can read, redistribute and modify the source code. As a result, open source software tends to evolve rapidly. People improve it, people adapt it and people fix bugs (Open Source Initiative, 2005). This usually happens at a speed that is faster than what could be achieved by commercial software development. Furthermore, Surman and Diceman (2004) suggest that with open source software there is less likelihood of becoming tied or dependent on a single software provider or being trapped into long-term software support contracts which restrict flexibility in implementation. Open source software also offers better security as more programmers are able to participate in the debugging of code. Open source software also usually adheres to standards-based technologies and is important for maintenance and upgrade purposes (McAllister, 2004).

However, there are some drawbacks in using open source software. Some packages do not have the same level of documentation, training and support resources as their common closed source equivalents (Caton, 2004). Making a major switch from proprietary software can be costly and sometimes the costs outweigh the benefits. Open source tools are also usually designed for professionals and are not as user friendly as some commercial products (Roberts, 2005). While costs can be less than that of the commercial software, they still exist, specifically around configuration and support. Sometimes these costs come to almost half as much as their commercial counterparts (Galli, 2004).

EVALUATION OF PORTAL SOFTWARE: A REVIEW

There have only been a few evaluative studies of portal software. The most recent was conducted by the Development Gateway Foundation, a non-profit organization which builds open source portal tools. The study compares open source portal packages (Jetspeed, uPortal, LifeRay, eXo) in terms of their features (Development Gateway, 2005) including content management, forums, calendars, chat, personalization, e-commerce support, adherence to technology standards, internationalization, search and system scalability. As the study aimed to determine only the total number of features available from each package, the method used involved assigning one point to each feature and calculating the sum. Using this approach, the study suggests that Liferay has the most features (13) followed by eXo (9), Jetspeed (8) and uPortal with the least (6). Although this method is rudimentary, it clearly distinguishes between the four software packages, if one assumes: 1) that all the features have equal importance and therefore should have the

same weighting and 2) that all features have been accounted for by the study's authors.

In a similar study on open source portals, Akram et al. (2005) selected the following portal frameworks for evaluation: Sakai 1.5, uPortal, Gridsphere, eXo platform, Liferay and Stringbeans. Features were identified and categorized so that they could be easily compared against each other. The categories were JSR-168 compliance, ease of installation, ease of use, documentation, support services, administration of portal, customization, free useful portlets, performance, security, technology use, portal features, server dependency and WSRP compliance. The evaluation method used was based on accumulative aggregation of scores obtained from the various categories. Results revealed that Liferay led with 58 points, followed by eXo (57). Stringbeans and Gridsphere were tied at 51 points, while uPortal had 49. Sakai was last at 35 points.

Browning (2003) also conducted an evaluative study of portals but using different products; namely uPortal, Oracle Portal, Microsoft Sharepoint and Cleverpath. Here, uPortal was the only open source portal. In this study the features to be evaluated were classified into a rubric (grading criteria) and further classified into sections which included front-end look-and-feel, security, personalization, support, standards and administration. Each of these was then further divided into sub-categories and a point scale was attached for the different scores that could be attained. Each sub-category was assigned an associated weight, which was used to produce a section score. Finally, the section scores were totalled to give an overall score. Results showed that uPortal led with the highest score of 150 followed by Oracle Portal with 143 points. Microsoft Sharepoint received 136 points while Cleverpath almost tied at 135.

THE CANDIDATE OPEN SOURCE PORTALS

In this study, three main criteria were used to select open source portal software packages for evaluation:

- the software should be relatively well known and commonly used, and this was inferred from the number of installations;
- they were evaluated in the literature at least once;
- the software must be available for download and installation at no cost via an open source licence to facilitate evaluation.

Four packages fulfilled the criteria and were selected for evaluation. They were Jetspeed 2, uPortal 1.0, Liferay 3.6 and eXo 1.

- **Jetspeed** is a Java-based open source enterprise information portal developed under the Apache

Software Foundation (<http://portals.apache.org/jetspeed-2/>) and released under the Apache License. A stated aim of Jetspeed is to allow an organization to build portal applications quickly and it includes tools for both portal developers and user interface designers. The portal supports a variety of access channels include Web browsers and WAP-enabled mobile devices, and has been implemented in many organizations worldwide.

- **uPortal** (<http://www.uportal.org/>) is developed by the Java in Administration Special Interest Group (JASIG), an international consortium of academic institutions. A key differentiating factor of uPortal is that it focuses on the requirements of higher education institutions and thus boasts of a large installation base of universities around the world. Nevertheless, the software contains features that are important in any portal and was therefore included in the evaluation.
- **Liferay Portal** (<http://www.liferay.com/web/guest/home>) is released under the MIT License which is similar to the Apache License, and has no licensing fees or reciprocity clauses. The software is designed for ease of use and offers out-of-the-box functionalities that claim to rival commercial portal solutions. These include blogs, wikis, message boards and RSS. Its installed base consists of many commercial organizations from various industries and countries.
- The **eXo Platform** portal framework is released under the popular GPL License, and like Liferay, also has a commercial version. The portal claims more than 100,000 downloads since its release and is implemented by Fortune 100 companies as well as others across the US and Europe. Strong content management and search capabilities as well as support for Web accessibility are some of its advertised main strengths (<http://www.exoplatform.org/company/faces/public/community>).

METHOD

Use of evaluation checklists

Checklists are commonly used for evaluation of software (Tergan, 1998). Their popularity is probably due to their apparent ease of use: by providing a structured list of relevant criteria for the evaluators to go through systematically and indicate whether a given product matches the criterion (Goh et al., 2006). Stufflebeam (2001) had also demonstrated the practicality of evaluation checklists by constructing checklists for various purposes, including those for evaluating software. Software evaluation checklists have also been designed to be used by practitioners (Rowley, 1993; Tolhurst, 1992).

A software evaluation checklist typically contains a list of items, usually questions, structured according to categories. Questions may be quantitative, for example the cost of the software, or qualitative, for example the

reputation of the vendor. For quantitative queries, the evaluator may be asked to indicate, according to a pre-determined rating scale, the extent to which a particular software feature is represented in a product. For example, a portal software being assessed may be found to allow the configuration of its layout to be modified directly without programming (full support), or indirectly via programming (partial support), or its layout cannot be modified at all (no support).

Evaluation criteria formulation

The evaluation was carried out using a customized checklist formulated on the basis of the defined objectives of the project as identified in the third report to the ASEF (Goh, 2005), as well as the literature review. Attempts were made to ensure that the features listed were comprehensive and representative of their respective categories. See Appendix A for details of the checklist.

The checklist was designed with these considerations in mind:

- it should be generic enough to cater for all the portals regardless of whether or not they are open source or closed, even though only open source portals will be evaluated;
- it should be unbiased in highlighting the strengths and weakness of each portal;
- it should focus on the technical features and functionalities, instead of usability issues;
- it does not have to be understandable to non-technical people as it is intended for internal, non-public use.

With these design considerations as a guide, the checklist was developed through the following steps.

Step 1: Evaluation categories

Based on the requirements of the project eight categories of features were devised:

- *Taxonomy*. In the context of our evaluative study, taxonomy refers to the hierarchical classification scheme used to organize content in the cultural domain for efficient and effective retrieval (Bruno and Richmond, 2003). The process of taxonomy building involves not just the creation of the classification scheme, but also the development of a controlled vocabulary, where a limited set of authorized terms is to be used in indexing documents. Portals should support the creation and editing of taxonomies, either manually or automatically (for example, from metadata) or by importing from an existing one. The created taxonomy should be exportable and be able to be used as a basis for the generation of one or more browsing structures (Lau and Goh, 2006).
- *Metadata*. Metadata is structured information that describes, explains, locates or otherwise makes it eas-

ier to retrieve, use or manage an information resource (Hodge, 2005). Metadata schemes are developed by domain experts as a formal approach to resource description. A portal should support the incorporation of metadata for its resources and follow known standards such as RSS (RDF Rich Site Summary), Dublin Core, MARC, EAD (Encoded Archival Description).

- *Search*. Without a proper search function, the user can only manually locate information by browsing through classified directories of content. A powerful search mechanism should cater for full-text, metadata and Boolean search. The user may also expect some supportive features, such as specifying the number of hits per page, selecting the type of information for each hit, or highlighting of the search words within the results (Curtis and Dormer, 2005).
- *Content management system (CMS)*. The role-based access, aggregation, security and personalization capabilities of a portal are greatly enhanced when it is implemented with a CMS that enables users to author, collaborate on and publish new content to the portal (White, 2002). The CMS should manage the entire life-cycle of content management by providing a complete range of workflow capabilities. These include support of core activities like content creation, content contribution, content delivery, content authorization, content publishing, content reporting, content delivery and content maintenance. The CMS should also support additional features like the assignment of designated access to the content based on identified roles, importing and exporting of contents in different formats and protocols, version control and built-in editors.
- *Customization and personalization*. The customization capability of a portal enables look-and-feel preferences to be specified at the end-user and group levels (Jasco, 1999). For example, items can be placed on a page in a flexible manner, different skins can be used to express different themes, fonts and colours can be specified. In addition, changes to the settings of the user interface may be saved, or users may be allowed to develop their own UI templates. Related to customization, personalization enables the creation of user accounts and subsequently user profiles so that each user can arrange and organize the features and information presented to him or her (Jasco, 1999). Context-sensitive information may be presented to users based on the role and responsibility assigned to their user profiles. Such user profiles may also contain access and security information for the configuration of user or group identities in order to present the right information to the user and group of users.
- *Collaboration and communication*. Collaboration and communication tools allow people to work together without geographic or time constraints. Examples are shared integrated workspaces, wikis, blogs, discussion

forums, instant messaging and event-based SMTP, SMS or 'pop-up' mechanisms (Fitcher, 2005).

- *Security.* A portal should provide a set of security features for user authentication and access control. Besides discretionary and role-based access control, password authentication and expiry, and login attempt restriction, logging of usage activities and generation of usage reports should also be provided (Messmer and Mears, 2002).
- *Interoperability.* In order to achieve a certain level of interoperability, a portal has to comply with standards like JSR 168 (Java portlet specification), OAI (Open Archives Initiative), Z39.50 (American National Standard for information retrieval), WSRP (Web Services for Remote Portlets), SOAP (Simple Object Access Protocol), WSDL (Web Service Definition Language), UDDI (Universal Description, Discovery and Integration), ODBC and JDBC, as well as provide application programming interfaces (APIs) for features like taxonomy, metadata, search, content management, customization, personalization, collaboration, communication and security. Interoperability greatly enhances the flexibility and power of a portal by enabling the assembly of components based on standard interfaces and services, reducing dependency on the features provided by a single portal (Smith, 2004).

The detailed checklist is shown in Appendix 1. Questions raised to address the various aspects of each category were reviewed to ensure that no two categories overlap. Otherwise, the same point will end up being evaluated twice, hence with an inappropriate increased weighting.

Step 2: Rating scheme

Ratings to each feature can be 'yes', 'indirect support' or 'no', which are assigned points of 1, 1/2 and 0 respectively. While the rating of 1 or 0 is straightforward in assessing the existence or non-existence of a particular feature, the half-point rating caters for cases that when a feature is available through additional programming or configuration. Ratings can also be annotated, serving as supporting evidence.

Step 3: Weighting scheme

Compared to the rating scheme, the weighting scheme is more complex, since it serves to differentiate and distinguish the importance of each portal feature. Initially, it was thought that weight assignment could be done at the category level. However, some categories were found to contain more features than others. As such, categories with more features will generally have less weight per feature due to the division of the weight across each feature. Another approach was to assign equal weights to every category, but this again will not

provide an accurate reflection of portal performance since categories with fewer features will have a misleading indication that its features are more important than those of other categories.

Recognizing that these schemes would not accurately reflect the importance of each feature, the weights were assigned at the feature level independent of its category. To further prevent a category from having a higher weight due to the number of features it contained, the number of features per category was limited to 16.

A weighting scale of 1 to 5 was then devised for each feature:

- 5 if it is a core (mandatory) feature;
- 4 if it is a secondary feature;
- 3 if it is a supportive feature, a feature for standardization or a security feature;
- 2 if it is meant for automation, integration or portability purposes;
- 1 if it is for aesthetic purposes only.

For example, under the 'Search' category, full-text searching is considered a core feature as it has to be available for any useful information retrieval, while metadata searching is considered a secondary feature as its coverage is part of full-text searching. Boolean searching simply enhances the search capability and is thus considered a supportive feature.

A bigger challenge was in assigning the higher weights as they appeared very similar at first glance, creating a level of ambiguity. A guiding principle was therefore developed to address this issue. A weight of 5 would represent features which are absolutely necessary or features which incorporate a collection of other features. A weight of 4 would include important but not mandatory features, usually falling within subsets or subcategories of features. Removing them would not compromise the core capabilities of a portal. A weight of 3 refers to features required to support the core and secondary features. For example, the ability to customize a layout's 'skin' to express different themes may be useful, but since this capability is not required in order for the layout to be customized, it is thus considered a supportive feature.

Use of similar weighting schemes have been reported in the literature, for example, in automating the selection of mathematical software for recommendation (Lucks and Galdwell, 1992) or, more recently, in the selection of simulation software (Tewoldeberhan and Bardonnet, 2002).

Step 4: Scoring

The score for each feature is simply computed as the rating multiplied by its weight. It ranges from a minimum of 0 (a 'no' answer, regardless of the weighting) to a maximum of 5.

Table 1. Comparison of features among four open source software packages

		Jetspeed	uPortal	LifeRay	eXo Platform
1	Taxonomy (max score: 26)	0	5	0	0
2	Metadata (max score: 31)	5.5	11	0	0
3	Search (max score: 15)	9.5	7.5	9.5	12.5
4	Content management (max score: 54)	54	48	51	48
5	Customization and personalization (max score: 42)	21	42	28	42
6	Collaboration and communication (max score: 43)	24	17.5	37	31
7	Security (max score 18)	16.5	13.5	15	12
8	Interoperability (max score: 32)	27	26	19	20
9	Total (max score: 261)	157.5	170.5	159.5	165.5

Portal evaluation

The four portal software packages were evaluated separately by three evaluators with information science and technology backgrounds using the same checklist. Each evaluator went through each feature in a particular category (e.g. taxonomy) to rate it as directly supported, indirectly supported or not available at all. Remarks were added against each item to elaborate on how it was supported. For example, a portal may have a built-in mechanism to directly support the feature, or the feature may be implemented using the portal's APIs.

Next, for each portal, the results obtained from the three evaluators were compared. Since the evaluation focused on the availability of technical features, rather than the more subjective aspects such as usability issues, results obtained by different evaluators should be identical, even though the remarks may be different. Discrepancies were eliminated by a combined re-assessment by all three evaluators. In the event that a conclusion could not be established based on an inspection of the portal or the documentation, the organization owning the portal was contacted via e-mail for clarification. In our study, discrepancies were encountered in less than 5 percent of the features, while less than 2 percent of the features required further clarification.

Finally, the scores for each category were computed by multiplying the ratings with the weights and summing them. The results are analysed in the next section.

FINDINGS

Table 1 shows the results obtained from the checklist evaluation of the four portals.

Taxonomy

With the exception of uPortal, taxonomy support is non-existent for the selected portals, as indicated in Row 1 of Table 1. In the case of uPortal, the functionalities are basic (which explains the score of 5 out of a total of 26), with only manual creation and editing of taxonomies. Creation through other means like importing or conversion via metadata is not available, and the created taxonomy cannot be exported.

Metadata

As with taxonomy, metadata support for Liferay and eXo is lacking. Jetspeed's APIs cater for an indirect approach to the incorporation of metadata and customization of metadata schemes, but the metadata only adopts the Dublin Core standard. uPortal is the best performer as seen from Row 2 of Table 1 although its score of 11 out of a total of 31 cannot be considered impressive. The portal supports the incorporation of metadata directly, and adopts the Dublin Core and RSS standards. XML and RDF formats are supported indirectly through its channels. However, none of the portals supports MARC, EAD, or OWL.

Search

Most of the portals fared well in this aspect (see Row 3 of Table 1), considering that the maximum possible score is 15. All portals support full-text search directly, with the exception of uPortal, which relies on Google for its full-text and Boolean search, as well as for presenting search results. eXo is the clear leader as it is the only portal with direct support for full-text, metadata and Boolean search, while Jetspeed and Liferay tie with similar indirect support for metadata and Boolean search via other search engines.

While main search features are available, aesthetic features are lacking. None of the portals allows for the customization of search results. While eXo enables the user to specify the number of hits per page, it has a minor limitation of not being able to highlight the search words in a result page, which most search engines (including portals that rely on search engines) support.

Content management system

Workflow management, which lies at the heart of a good content management system, involves activities like content creation, contribution, review, authorization, publication, reporting, delivery and maintenance (White, 2002). All four portals scored well in this category (see Row 4 of Table 1) as they implement core workflow activities through their integrated CMS engines. In fact, Jetspeed managed a perfect score of 54 out of 54.

In addition, all four portals feature role-based access, which is the assignment of access rights to the content based on identified roles. Content can also be imported and exported in different formats (e.g. XML, text, CSV) and using different protocols (e.g. ODBC, JDBC) for all four portals. Liferay and Jetspeed scored slightly better compared to uPortal or eXo, as they come with built-in editors for modifying text documents. Furthermore, Jetspeed is the only portal that incorporates a version control mechanism.

Customization and personalization

uPortal and eXo boast comprehensive support for both customization and personalization. They fulfilled all the requirements in the checklist, obtaining the maximum possible score of 42 (see Row 5 of Table 1). In terms of customizability, the layout of both portals enables flexible placement of items on a page, different 'skins' of the same layout can be used to express different themes, fonts and colors can be selected, and their GUI panes can be moved or resized. All these settings can be saved and users can even develop their own UI templates. In terms of personalization, user accounts and profiles can be created so that each user can arrange and organize his or her information. Context-sensitive information may also be presented to each user based on the role and responsibility assigned to his or her profile. For example, if a user has indicated an interest in technology-related information in the profile, such information

will be displayed when available. Both individual and group identities may also be assigned based on access and security information defined in the profiles, in order to present appropriate information to the individual or group of users.

In comparison, Liferay and Jetspeed are less impressive with scores about half those of uPortal and eXo. Liferay allows user accounts to be created, but not user profiles or individual and group identities. Unlike Liferay, Jetspeed does not even let users select the type of content they want, making it the worst performer in this category. Creation of user profiles and subsequent configuration of individual and group identities are possible, but only indirectly through APIs. In contrast to personalization, the only weakness in terms of customization is that users cannot create their own UI templates.

Collaboration and communication

The difference in performance between the portals in this category is distinct, with a variation in score of around 6.5 between any two portals. Liferay was the best performer with a score of 37 out of 43. This is attributable to the availability of five important instruments of collaboration and communication, namely, shared integrated workspaces (for uploading/downloading of documents, real-time discussions and chat sessions, e-mail exchanges, posting of messages, real-time video conferencing, desktop sharing sessions and so on), wikis, blogs, discussion forums and instant messaging. Liferay is the only portal that directly supports the creation of all five mechanisms. eXo follows with indirect support (via programming) for blog creation and direct support for the rest. Jetspeed offers support for all five mechanisms, but only indirectly. uPortal is slightly worse, without support for wikis. In terms of less important features, Liferay excels in event-based alerting, followed by eXo, while Jetspeed is strong in online calendar functions (see Row 6 of Table 1).

Security

Jetspeed is the best performer in this category (scoring 16.5 out of 18), as a result of its direct support for almost all features, ranging from password authentication and expiry, restrictive login attempts, discretionary access control, role-based access control, to usage activity logging (see Row 7 of Table 1). Only usage report generation is weak, with output in XML format as the only possibility. This feature is directly supported in Liferay, which also comes with other features similar to what Jetspeed is capable of, except that it has no way to restrict login attempts. uPortal directly supports half the number of features while it indirectly supports the other half, namely role-based access control, usage activity logging and usage report generation. eXo's last position is due to its lack of discretionary access control and difficulty in implementing restrictive login attempts and password authentication and expiry.

Interoperability

Row 8 of Table 1 shows that Jetspeed and uPortal are more interoperable than the other two portals (with scores close to the maximum possible of 32), because of the presence of support for JSR 168, WSRP, SOAP, WSDL and JDBC standards, as well as the provision of APIs for taxonomy, metadata, search, content management, security, customization, personalization, collaboration and communication. Only OAI and ODBC standards are not available. uPortal scored a point less due to its indirect UDDI support. Liferay and eXo platform lose out mainly due to the lack of any standards-based support for taxonomy, metadata, search, customization and personalization.

Individual portal profiles

Having examined the eight categories as shown row-wise in Table 1, the discussion now centres on individual portals by examining Table 1 column-wise. Jetspeed's strengths lie in the areas of content management system, security and interoperability. uPortal's strengths lie in content management, customization and personalization, and interoperability. Liferay excels in content management, collaboration and communication, and security. Like Liferay, eXo performs well in content management, customization and personalization, and search. The scores obtained by the portals in each area were summed to compute an overall score. As shown in Row 9 in Table 1, uPortal emerges as the overall best performer with a total score of 170.5, followed closely by eXo, Liferay and Jetspeed, with total scores of 165.5, 159.5 and 157.5 respectively.

It can be seen that uPortal is the only open source portal software package that provides features for all eight categories identified in the study: taxonomy, metadata, search, content management, customization and personalization, collaboration and communication, security and interoperability. uPortal performed reasonably well across all the eight categories, especially in the area of taxonomy. Here the other portals fell short with none supporting these features. Metadata support was also strong for uPortal, with Jetspeed tailing very close behind.

It appears therefore that uPortal is a strong candidate for the ASEF cultural information portal in that it has the potential to deliver most of the important functions identified. In addition, uPortal has the ability to customize portal layout and aesthetics, and the ability to personalize the portal according to individual or group profiles. It is also interoperable with many other external systems by conforming to widely accepted standards.

Although eXo did not support taxonomy and metadata, it is very strong in all other aspects, especially in the areas of content management, customization and personalization. It also scored well for collaboration and communication, and has stronger search features compared to uPortal. With many of the features built in, eXo allows implementers to roll out a portal with ease.

Liferay lacked support in the taxonomy and metadata categories. Instead, Liferay excelled in the areas of content management, and collaboration and communication. In fact, of all the portals, it best supports the latter category. Liferay also has many advantages over the other portals in that it offers out-of-the-box portlets for various features, facilitating easier deployment. A portal implemented using Liferay can therefore be set up in only a few days and include support for blogs, calendar, CMS, image gallery, mail, message boards, polls, RSS feeds and even wikis. Another distinct advantage is its multilingual support, including Chinese, English, French, German, Japanese, Korean and so on. This would be particularly useful in the case of ASEF where communication between cultures is emphasized.

Although Jetspeed has the lowest score from the study, it has several strengths. Jetspeed is developed by the Apache Software Foundation, which gives it an advantage of strong branding and support among open source developers and users, as well as access to tools developed under Apache which enhances functionality and ease of development. Examples include Bridges, Pluto and Graffito. Bridges provides support for JSR-168 compliant portlet development using common web frameworks, while Pluto serves as portlet container that offers developers a working example platform from which they can test their portlets. Finally, Graffito supports the development of content-based applications such as CMS, forums, blogs and so on (Apache Portals, 2005).

In addition, from the previous figures, Jetspeed is the best performer for content management, obtaining a score of 54. It also has support for features pertaining to metadata, whereas Liferay and eXo platform had none. It is also the best performer in the field of interoperability and the highest score for security which is crucial in today's Internet environment.

CONCLUSION

The present work represents a first step in the development of an exhaustive evaluation tool for portals. The main strength of the tool lies in its generalizable application on both open source as well as commercially developed software. By featuring eight categories of technical features and functionalities, it affords an unbiased appraisal of each portal software under consideration.

Strictly based on the results of this evaluation, we can conclude that uPortal is the most comprehensive open source portal software available. However, as good as uPortal is, a fair amount of development is still required to provide features that are not supported directly. Furthermore, each of the four portals has unique strengths and weaknesses, and each may be the best fit under different circumstances and needs, although we feel that more work is needed to improve all of these packages.

There are still gaps in our evaluative framework. To begin with, the focus is largely on software features. Not considered are factors such as hardware, time, manpower, money and other resources, as these may vary depending on the implementing organization or individual. Also, the weights assigned to the various categories are based only on the informed opinions of the three evaluators. It would be useful to elicit feedback from other portal developers and users to confirm the weights used. In addition, the evaluation process was based on an 'as far as we can find' approach, using a test portal installation and documentation from various sources. Therefore there is a possibility that a few assumptions about features that are not supported may be wrong. It would be a valuable addition to the study if actual live portals could have been tested as well. Finally, other open source portals such as Digijava, Sakai, Gridsphere and Stringbeans and so on could also be evaluated using the checklist in the future.

REFERENCES

- Akram, A., Chohan, D., Wang, D., Yang, X. and Allan, R. (2005) 'A Service Oriented Architecture for Portals Using Portlets', Proceedings of the 2005 UK e-Science All Hands Meeting, 19–22 September, URL (consulted October 2005): <http://www.allhands.org.uk/2005/proceedings/papers/406.pdf>
- Apache Portals (2005) URL (consulted October 2005): <http://portals.apache.org/>
- ASEF (2005) *Cultural Partnership Mapping*. Singapore: Asia-Europe Foundation.
- Browning, P. (2003) *Portal Evaluation. Report*. Bristol: University of Bristol.
- Bruno, D. and Richmond, H. (2003) 'The Truth about Taxonomies', *Information Management Journal* 37 (2): 44–51.
- Butters, G. (2003) 'What Features in a Portal?', *Ariadne* 35, URL (consulted October 2005): <http://www.ariadne.ac.uk/issue35/butters/>
- Caton, M. (2004), 'Sugar Sale: Just the SFA Basics', *eWeek* 21 (50): 48.
- Curtis, A. and Dormer, D.G. (2005) 'Why Federated Search?', *Knowledge Quest* 33 (3): 35–7.
- Development Gateway (2005) URL (consulted October 2005): <http://www.developmentgateway.org>
- Fisher, R., and Varbanova, L. (2005) 'Research on Cultural Information Sources in Europe', in *Cultural Partnership Mapping*, pp. 13–83. Singapore: Asia-Europe Foundation.
- Fitcher, D. (2005) 'The Many Forms of E-collaboration: Blogs, Wikis, Portals, Groupware, Discussion Boards, and Instant Messaging', *Online* 29 (4): 48–50.
- Galli, P. (2004) 'Reverse Migration: From Linux to Windows', *eWeek* 12 (39): 11–12.
- Goh, D. (2005) 'Research on Technical and Implementation Aspects of Portal Development', in *Cultural Partnership Mapping*, pp. 158–85. Singapore: Asia-Europe Foundation.
- Goh, D.H., Chua, A., Khoo, D.A., Khoo, E.B., Mak, E.B. and Ng, M.W. (2006) 'A Checklist for Evaluating Open Source Digital Library Software', *Online Information Review* 30 (4): 360–79.
- Hodge, H. (2005) 'Metadata for Electronic Information Resources: From Variety to Interoperability', *Information Services and Use* 25 (1): 35–45.
- Jasco, P. (1999) 'Rise of the Personalized Web Portals', *Information Today* 16 (7): 25–6.
- Keller, A. (2001) *I-Assure*. Portal market and vendor product report, National Communications System, USA, URL (consulted October 2005): http://www.ncs.gov/informationportal/Web_Portal_Report.doc
- Lau, E.P. and Goh, D.H. (2006) 'In Search of Query Patterns: A Case Study of a University OPAC', *Information Processing and Management* 42 (5): 1316–29.
- Lucks, M. and Gladwell, I. (1992) 'Automated Selection of Mathematical Software', *ACM Transactions on Mathematical Software* 18 (1): 11–34.
- McAllister, N. (2004) 'Opening up the Code', *InfoWorld* 26 (49): 42–8.
- Mack, R., Ravin, Y. and Byrd, R.J. (2001) 'Knowledge Portals and the Emerging Digital Knowledge Work Place', *IBM Systems Journal* 40 (4): 925–55.
- Messmer, E. and Mears, J. (2002) 'Web Portals Pose Security Challenge', *Network World* 19 (2): 1–2.
- Open Source Initiative (2005) *The Open Source Definition*, URL (consulted October 2005): <http://www.opensource.org/docs/definition.php>
- Roberts, G. (2005) 'Linux Adventures on a Laptop', *Computers in Libraries* 25 (5): 33–5.
- Rowley, J.E. (1993) 'Selection and Evaluation of Software', *AsLib Proceedings* 45 (3): 77–81.
- Smith, M.A. (2004) 'Portals: Towards an Application Framework for Interoperability', *Communications of the ACM* 47 (10): 93–7.
- Stufflebeam, D.L. (2001) 'Evaluation Checklists: Practical Tools for Guiding and Judging Evaluations', *American Journal of Evaluation* 22 (1): 71–9.
- Surman, M. and Diceman, J. (2004) *Choosing Open Source: A Guide for Civil Society Organizations*. The Commons Group for the Association for Progressive

- Communications (APC), USA, URL (consulted October 2005): <http://www.commonsc.ca/articles/fulltext.shtml?x=335>
- Tergan, S.O. (1998) 'Checklists for the Evaluation of Educational Software', *Innovations in Education & Training International* 35 (1): 9–12.
- Tewoldeberhan, T.W. and Bardonnnet, G. (2002) 'An Evaluation and Selection Methodology for Discrete-event Simulation Software', *Proceedings of the 2002 Winter Simulation Conference*, pp. 67–75.
- Tolhurst, D. (1992) 'A Checklist for Evaluating Content-based Hypertext Computer Software', *Educational Technology* 32 (3): 17–21.
- von Hippel, E. and von Krogh, G. (2003) 'Open Source Software Development and the Private-collective Innovation Model: Issues for Organization Science', *Organization Science* 14 (2): 208–23.
- White, M. (2002) 'Managing Content Management System Selection', *EContent* 25 (7): 42–3.

APPENDIX 1

TAXONOMY

- Does the software support creating and editing of taxonomies?
- Is taxonomy creation automatic?
- Does the software support importing an existing taxonomy?
- Is the taxonomy created exportable?
- Can browsing structures be created based on the taxonomy?
- Can the taxonomies be created based on the resources' metadata?
- Can we select which metadata fields to construct the taxonomy from?

METADATA

- Does the software support the incorporation of metadata for its resources?
- Does the metadata follow RSS (RDF Rich Site Summary) Standard?
- Does the metadata follow Dublin Core Standard?
- Does the metadata follow MARC Standard?
- Does the metadata follow EAD (Encoded Archival Description) Standard?
- Does the representation of the metadata follow XML (Extensible Markup Language) formats?
- Does the representation of the metadata follow RDF (Resource Description Framework) formats?
- Does the representation of the metadata follow OWL (Web Ontology Language) formats?
- Are the metadata schemes customizable?
- Is metadata creation automatic?

SEARCH

- Does the software support full text search?
- Does the software support metadata search?
- Does the software support Boolean search?
- Does the software allow specification of the number of hits per page?
- Does the software allow specification of the type of information shown for each hit?
- Does the software highlight the search words in the result page?

CONTENT MANAGEMENT SYSTEM

- Does it support content creation?
- Does it support content contributing?
- Does it support content reviewing?
- Does it support content authorization?
- Does it support content publishing?
- Does it support content reporting?
- Does it support content delivery?
- Does it support content maintenance?
- Is assignment of access rights to the content based on identified roles supported?
- Is importing and exporting of contents in different formats and protocols supported?
- Is version control supported?
- Does it have built-in editors?

CUSTOMIZATION AND PERSONALIZATION

- Can the layout be customized?
- Can changes to the settings of the user interface be saved?
- Can the 'skin' of the layout be customized?
- Can colors and shades be specified?
- Can fonts be specified?
- Are the GUI panes resizable and movable?
- Does it allow users to develop their own user interface templates?
- Can users specify the types of content they want?
- Does it support the creation of user accounts?
- Does it support the creation of user profiles?

Does it support configuration of individual identities based on user profiles?
Does it support configuration of group identities based on user profiles?

COLLABORATION AND COMMUNICATION

Can Wikis be created?
Can shared integrated workspaces be created?
Can Blogs be created?
Can discussion forums be created?
Is there support for instant messaging?
Is there any event-based SMTP alerting mechanism?
Is there any event-based SMS alerting mechanism?
Is there any event-based 'Pop-up' alerting mechanism?
Is there support for expertise locators?
Is there support for resource ratings by users?
Is there support for commenting of resources by portal users?
Is there support for online calendaring?
Is there ability to see who's logged into the portal?

SECURITY

Is there any way to implement password authentication and expiry?
Is there any way to restrict login attempts?
Does it provide a mechanism to allow discretionary access control?
Does it provide a mechanism to allow role-based access control?
Is there any facility to log usage activity?
Does it provide a mechanism to generate usage reports?

INTEROPERABILITY

Does it support JSR 168?
Does it support OAI?
Does it support Z39.50?
Does it support WSRP?
Does it support SOAP?
Does it support WSDL?
Does it support UDDI?
Does it support ODBC?
Does it support JDBC?
Does the portal provide APIs for taxonomy?
Does the portal provide APIs for metadata?
Does the portal provide APIs for search features?
Does the portal provide APIs for content management?
Does the portal provide APIs for customization and personalization?
Does the portal provide APIs for collaboration and communication?
Does the portal provide APIs for security?