Facilitating Inter-organisational Collaboration via Flexible Sharing of Rapidly Developed Web Applications

Ioakim (Makis) Marmaridis*, Xufeng (Danny) Liang, and Athula Ginige

School of Computing and Mathematics, University of Western Sydney
Sydney, Australia
makis@scm.uws.edu.au,
danny@scm.uws.edu.au,
a.ginige@uws.edu.au
http://www.uws.edu.au

Abstract. Increased competitiveness in business requires organisations to work with each other to extend their reach. To enable flexible and nearly ad-hoc information sharing and collaboration we have advocated the use of Dynamic eCollaboration. There are three types of information that must be shared: publicly accessible, information provided via online services, and finally, parts of or the entire web application. To facilitate the rapid pace of Dynamic eCollaboration, web applications must be created quickly and the sharing mechanism should be driven by end-users. In this paper we are presenting the integration of the Bitlet framework for web-based information sharing with our work on the Smart Business Object (SBO).

1 Introduction

Information sharing on the web is one of the cornerstones of eCollaboration [1,2] and the need for it is increasing as the growth in this area of research indicates [3,4,5,6]. To that end, many approaches and tools already exist for sharing information on the web. Unfortunately, existing approaches and tools are limited and unable to cater for anything beyond the sharing of publicly accessible information. In practice, there is a strong need for people to be able to share information that is not always publicly available, collaborate using web applications jointly, and easily aggregate the shared information. These are key features of information sharing that eCollaboration depends on.

As eCollaboration is gaining popularity as a business practice, it is necessary for information sharing approaches to keep up with the user requirements. At the same time, it is not uncommon to require applications that are purposely built to support collaborative work. These application must also support not only the specifics of the business logic, but also collaborative access and work distribution. In this paper, we showcase the combination of our Bitlets with

* Corresponding author.

J. Yang et al. (Eds.): UNISCON 2009, LNBIP 20, pp. 53–64, 2009.

the Smart Business Object (SBO). The Bitlet framework [7] facilitates sharing of web-accessible information including parts, or entire, web applications. The Smart Business Object [8] framework caters for rapid creation of web applications by end-users. The combination shows how applications can be created very quickly and get jointly used in the context of collaboration between different partners whose staff are geographically separated. Also in this paper, we present a walk-through example of an application that we have built and used recently in sharing support material between a number of researchers at our and a collaborating institution.

2 Need for Sharing Web Applications

Throughout our research in the area of flexible organisational eCollaboration that we also refer to as Dynamic eCollaboration, it is shown that collaboration is a key ingredient to organisational success [9], [10]. The ability to perform Dynamic eCollaboration is limited in part by how well businesses can share information with one another. The degree of sharing is in turn constrained by current approaches to information sharing. We therefore argue that a new approach to information sharing can introduce additional capabilities for organisations subsequently facilitating Dynamic eCollaboration better. To that end, we established through our research experience a set of key features that information sharing in the context of Dynamic eCollaboration must possess. We then reviewed the different approaches to information sharing against those features looking at the gaps that exist. Finally, building upon that knowledge, we created a new approach for information sharing on the web, based on Bitlets, which incorporates the key features identified previously. Now through this publication we extend the toolset available to organisations practising Dynamic eCollaboration via allowing not only sharing of existing web based applications but also the rapid creation of new ones as well if required.

2.1 Features of Information Sharing in Dynamic eCollaboration

In our experience to date, effective information sharing within the context of Dynamic eCollaboration must offer users control, flexibility and choice. In the combination of Bitlets with SBO we strive to maintain these abilities and continue empowering the end-users. The table below summarises the list of features we see as necessary for effective information sharing in Dynamic eCollaboration.

2.2 General Principles of Sharing Information on the World Wide Web

The most widely exploited feature of the World Wide Web when it comes to information sharing is that each element found online (also known as a resource) can be uniquely identified and accessed via a URL (universal resource location). These URLs were originally devised in order to allow for resources to link to each other (as in page to page links) and also to allow for other types of resources such

Table 1. Features of Information Sharing in Dynamic eCollaboration

Control	Users must be able to initiate sharing of information with one or more other users and maintain the ability to revoke access to that information or amend the terms of sharing at any time.
Flexibility	The information to be shared may come be publicly accessible, accessible as an online service via authentication or directly from an existing or a new, user-created web based application. Users must be able to capture and share information from any of these sources.
Choice	Users must be able to share information that is static or that changes over time. Changing information should be statically, or dynamically.

as images and rich media to be included into pages. Fortunately, this properly of the web can also be exploited when it comes to information sharing. At the most primitive level information sharing can be achieved by simply sharing the URL of a particular resource with others.

Although all sharing methods presented in this paper, including Bitlets, fundamentally rely on URLs for addressing web resources, some of the sharing approaches also allow for additional capabilities such as metadata descriptions, annotations, sharing with context, information aggregation and access arbitration to shared information just to name a few. In our previously published work we have performed a comparative analysis between existing approaches to information sharing and Bitlets [7].

2.3 Limitations of Current Approaches to Sharing

All the current tools and approaches that we reviewed during the course of our research follow the same general architecture pattern shown in Figure 1.

Under this pattern, a user discovers some interesting information on the web and decides to share it. The URL of the page containing that information will then need to be communicated via one of many available methods. The second user is then able to browse to that URL themselves. Upon their first visit to that original web page, they can then bookmark that URL for future reference thus completely bypassing the mechanism that was used to communicate the URL to them in the first place. Also, once the first user has shared the URL to the page he has found to be of interest, he has no ability to revoke this. Depending on the tool he used to communicate the URL he may be able to revoke the notification message but other users may still have the original site bookmarked.

Architecture aside, most of the current tools do not allow for information to be shared maintaining the context of the page it came from intact. From all the solutions reviewed, only copying and pasting from the browser into a document allows for context to be partially maintained (it does not always work as expected and success is dependent on particular browser versions and word processing applications used). A handful of online sharing services exist as well to facilitate inter-personal sharing however even the more elaborate ones also allow for the information page to be captured and stored on their systems however they

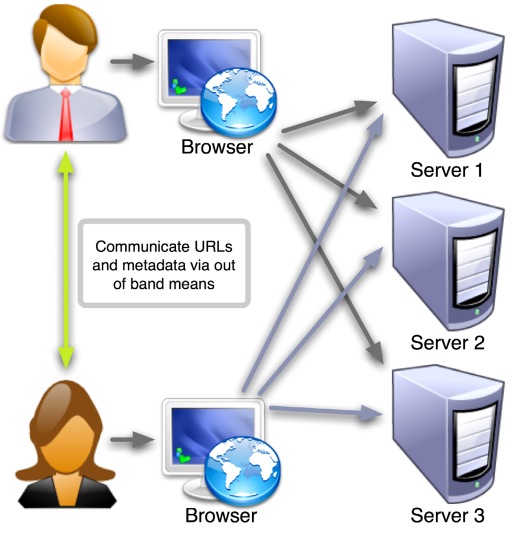

Fig. 1. Generalised Architecture Pattern of Current Information Sharing Approaches

only capture the html content of the page. They leave links to included resources such as images, CSS stylesheets and javascripts unaffected pointing back to the originating site. Therefore, images may still change or even completely disappear overtime. CSS stylesheets may also change to reflect a new site design for instance endangering the readability of the cached html copy. Bitlets come to address all of the limitations above as well as other smaller ones such as allowing different versions of the shared information to be kept if required.

3 Bitlet Framework Overview

3.1 Bitlet Fundamental Architecture Shift

We define a Bitlet as being a stand-alone, sharable bundle of information and associated metadata (such as user provided description and keywords or tags) that can be shared with others over the web. Bitlets encapsulate the entire content of a web page including any referenced resources such as images, media files, style sheets or JavaScript files. Bitlets are designed to be end-user driven and require very few, simple steps to create and work with. Unlike passing a single URL or a URL with a description to another person, a Bitlet encapsulates the page content and other metadata. A Bitlet also offers increased flexibility when it comes to sharing it with others compared to traditional means of sharing. With current methods for information sharing, one typically has to use out of band methods of sharing URLs, documents or other types of pointers to web accessible data. On the other hand, a Bitlet can be selectively shared with individuals or groups of individuals and its creator can at any time revoke access to it. This ability alone offers a great amount of flexibility for end-users and will

be another great motivation for them to use Bitlets where they would normally rely on more traditional sharing methods.

Sharing using Bitlets is fundamentally different from the way all other current approaches operate. This is because Bitlets use a different sharing paradigm. Instead of sharing a pointer (URL) to information residing on a web page somewhere, the entire information and descriptions, keywords and other metadata is shared instead. All access to this Bitlet goes through a single logical path (via the Bitlet Server) and all refreshing of the Bitlet content or interactive use of that content again goes through the same logical path and it is arbitrated. Figure 2 shows a view of the Bitlet architecture.

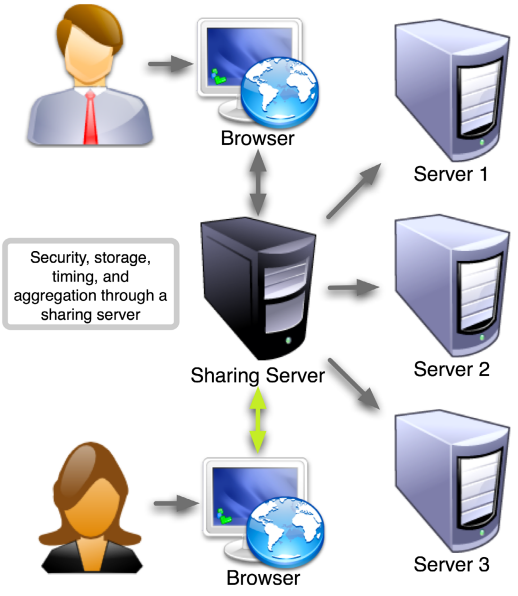

Fig. 2. Bitlet Architecture that Allows Control of Sharing, Aggregation and Two-way Sharing

The key idea here is to arbitrate access to the actual web accessible data by means of the Sharing or Bitlet server. Instead of making the shared data readily available by directly pointing users to it, a Bitlet can instead offer access to the same data, enrich the user experience by offering metadata about the shared data and allow the Bitlet creator control over the sharing throughout the collaboration process. The Sharing or Bitlet server need not be a single physical computer either. It can be distributed across a number of servers for better redundancy and fault-tolerance. Where multiple Bitlet servers are used, they have the ability to communicate with one another and transparently synchronise selected Bitlets for later disconnected (or offline) use.

3.2 Key Advantages and Features

Because of the very different architecture that Bitlet based sharing adopts in comparison to current approaches it is possible to offer a whole host of unique features that other methods cannot match. These features include: Sharing of pages that require authentication - via the Bitlet server impresonating access. Ability to share content interactively (a web form for instance). Maintain full access control of the shared information using role based access control (RBAC) [11], [12]. Share information that is automatically refreshed when required and cache a copy locally. Group multiple Bitlets together into a bundle and present or manage them as a whole.

3.3 The Role of Bitlets in Facilitating Dynamic eCollaboration

Bitlets have been created as a novel method for sharing information in the context of Dynamic eCollaboration, however no amount of technology will be successful in assisting in the adoption of Dynamic eCollaboration unless end-users actively embrace it and make use of it. We strongly believe that Bitlets are an effective method for sharing information on the web and that end-uses will be highly motivated to use this over other existing methods. We base this belief in the solid analysis of requirements that we have performed in previous work about what Dynamic eCollaboration needs to be effectively supported and made accessible to organisations of all sizes and its end-users. While we have not yet had the chance to carry out a longitudinal study of how people adopt to the use of Bitlets, there are several key advantages Bitlets have to offer making them quite attractive. These are as follows:

- Trust levels can be better expressed through varying the degree and amount of sharing quickly and easily. All sharing is done from inside the browser. There is no need other system such as email, phone or instant messaging for carrying the sharing metadata out of band.
- Enablement of users to share useful information with others only for the duration of a collaborative project and maintain complete control of the sharing. Unlike sending an email with a URL that once sent is nearly impossible to revoke and even if revoked the recipient still has access to the information since they are accessing it directly via the provided URL.
- Internal systems remain better secured through sharing parts of them via Bitlets. Before Bitlets were available it was not uncommon for people to have to create temporary guest accounts on internal systems of theirs in order to allow collaborating partners to access data from them. With the Bitlet server arbitrating all access to web applications such requirement is waived. Selective pages of a web application can be shared without the creation of new users or security roles in the web application.
- Convenience and speed of setup and use is another key motivating factor we expect to attract users to the use of Bitlets as the preferred sharing method for collaboration. Having to identify upfront what information a business

partner requires and then do the necessary changes to existing web applications and data as to allow such access is very hard to do. Particularly so when collaboration is of short to medium term and information sharing requirements change frequently during the project's lifetime. Bitlets allow end-users to share information as they see fit. No expert IT involvement is necessary in re-configuring existing systems or managing security permissions etc.

Empirically, we are seeing a lot of excitement from our Bitlet users and its increased capabilities in facilitating Dynamic eCollaboration.

4 Advantages of Using SBO for Building Web Applications

When two or more organisations decide to engage in Dynamic eCollaboration, it is not uncommon that their requirements for the project at hand are rather unique and they could require a specialised application to be created in order to support those requirements. It is for these cases where extremely rapid development of simple line of business web applications is needed. The Smart Business Object [8] fills this need.

Smart Business Object is a realisation of the End-User Development (EUD) paradigm (see [13]) for making software "easy to develop". Using the Smart Business Object framework, data driven web applications can be created in two steps:

1. Model the necessary business objects using SBOML (Smart Business Object Modelling Language), a compact, textual modelling language with near-English syntax
2. Generate different web views of the modelled business objects using the SBO's UI Generation Tool

Modelling Smart Business Objects using SBOML means the generation of the underlying database(s) for object persistence, the ORM (Object Relational Mapping) mapping classes, and a rich portfolio of web user interfaces for presenting the modelled business objects onto the web.

Thus, functional web applications can be easily generated within an extremely condensed timeframe. SBOML has a high-level of abstraction and is based on English lexicon. To model an "employee" business object, for instance, we can literally type the definition (in SBOML expressions) into the SBO Builder tool as shown in Figure 3. In runtime, Smart Business Object is capable of making logical inferences for the most suitable web user interfaces to present the business object's attributes. For example, SBO can automatically renders a Google map for an *address* attribute and displays a *photo* attribute as image (as shown in Figure 4).

Once the business objects are modelled, we can request the rendering of commonly used web *UI Components* in business web applications, such as tables, forms, and charts, for the modelled business objects using the *UI Generation*

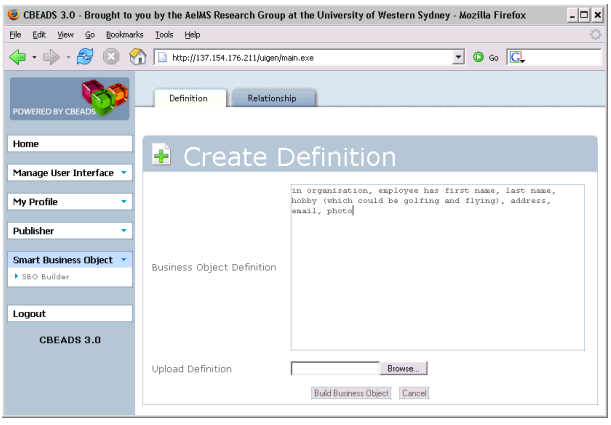

Fig. 3. The SBO Builder

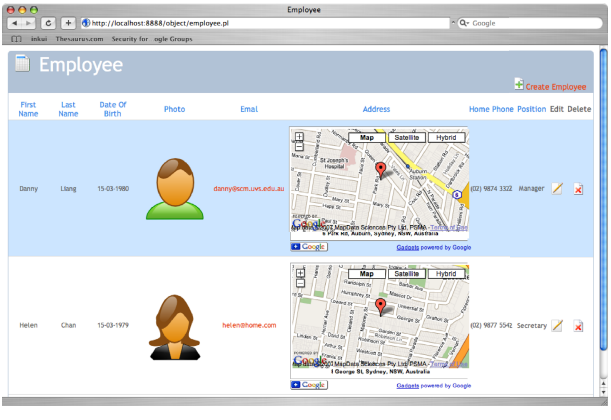

Fig. 4. A Generated Web Application

Tool. The *UI Generation Tool* provides a rich set of options for fine-tuning the generated web user interfaces. Figure 4 is an sample application generated using the toolkit. Consequently, an adept business user with some experience in interacting with web applications can be trained quickly to use the SBO's toolkit to create *instant* web applications.

5 Sharing SBO Based Applications Using Bitlets

These core advantages of combining the Bitlet framework for sharing with SBO is that should new applications need to be created, they do not need to pay any attention to sharing or make any special provisions for supporting collaboration. Instead, they can be built as stand-alone web applications that with the integration of Bitlets can be instantly turned into collaborative applications. Therefore

they can be shared across the network with one or more business partners as it is required. Also the Bitlet framework allows the sharing of web applications and collaboration using those even when these do not exist before hand on each business partner. Only one of the business partners must have the application and they can create bitlets out of it and share those. Bitlets are self-standing and contain all the information required for collaborative use, including the ability to post information back to the originating application or periodically refresh the content of parts of web applications that have been shared.

To showcase the power of integrating Bitlets with SBO-created applications we are describing in this section a recent application that we created for managing research resources of interest between a disparate group of researchers situated in physically different locations. The motivating need was to allow sharing of research material amongst a group of researchers quickly and easily using the web as a medium.

Using this system, researchers that wish to share information with others in their group access CBEADS [14] - a web based framework for developing and deploying web applications - and once they authenticate against it, they can access the Regina environment which is the SBO built appplication.s server.

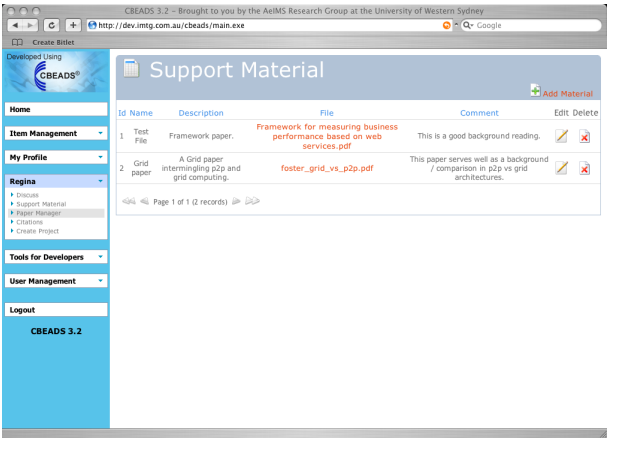

Fig. 5. Main Application Screen - Showing all available research resource materials

This is the main application page and here one can see what resources are available, can also add a new piece of material, edit existing ones or remove selected ones.

To share the uploaded materials, the user can click on the special bookmarklet with the name "Create Bitlet". This will bring up the details page for creating a new Bitlet and the user provides additional details as shown in Figure 6.

The Mode setting defines whether the Bitlet contents are going to be changing over time or not. Static Bitlets are like a snapshot as at the time they were

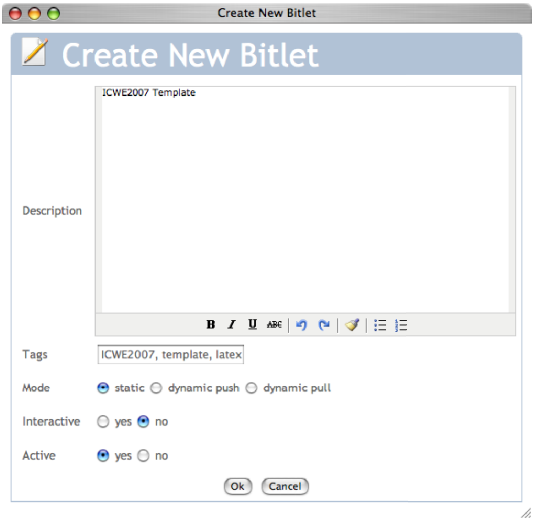

Fig. 6. Create a Bitlet - Make a bitlet out of the main screen of the application

created. Dynamic push means that the Bitlet get automatically updated at regular intervals. Dynamic pull means that the recipient will trigger a Bitlet update every time they open it up. The Interactive setting determines whether a Bitlet allows the recipient to interact with it or not. Non-interactive Bitlets do not allow form post-backs whereas interactive ones do. Without going into implementation details, if the "add material" form was shared as an interactive Bitlet, recipients of it could then fill the form and submit it so that any new materials would be added in the original application.

Once the Bitlet has been created it will automatically become accessible to others in the group who will be able to access the original application through it without having direct access to the application themselves.

Bitlets are agnostic of their content, they can be used to capture publicly available web pages or other web accessible content as long as it is accessible by URL. SBO-created applications are well suited for interacting with Bitlets. SBO applications allow for each and every page that is generated to be addressable via a unique URL, therefore the Bitlet framework can easily distinguish pages and retrieve them in case they are dynamically shared.

Where authentication is required the Bitlet framework is able to perform user impersonation and handle this requirement transparently from the end-user and recipient of the Bitlet. Also, when Bitlets content needs refreshing the Bitlet framework handles this transparently as well.

6 Conclusion

In this paper we have outlined the need for end-user driven, flexible and nearly ad-hoc collaboration via sharing of web applications. We also presented the

increased power that comes from combining Bitlets with Smart Business Object, two complimentary technology frameworks for sharing and rapidly building web applications that fulfil collaboration objectives.

Via the integration of our existing technologies of the Bitlet framework for web information sharing and the Smart Business Object, we have been able to provide an environment for rapid web application creation and sharing that is end-user driven and extensible.

These two key technologies provide a great foundation for collaboration however a lot more is needed for providing end-to-end Dynamic eCollaboration. This paper has not covered some of these additional topics such as virtual teams and issues of trust between collaborating partners. These are covered in some of our work previous published [1,2,15].

Although what we have presented in this paper offers a complete solution for sharing web based applications between individuals, many interesting areas of research and future work remain available and should be worked on. On the sharing front, Bitlets should eventually be extended to incorporate better aggregation of Bitlets allowing the sharing of information that is derived from a set of Bitlets. Also content tagging and highlighting should be supported within each Bitlet rather than treating them as small black boxes once shared. On the other hand, Smart Business Object should better faciliate the aggregation of business objects in heterogeneous forms, such as those that exist in service-oriented architecture, while providing consistent web user interfaces for transparent access to the aggregated business objects.

References

1. Marmaridis, I., Ginige, J., Ginige, A.: Web based architecture for dynamic ecollaborative work. In: International Conference on Software Engineering and Knowledge Engineering (2004)
2. Marmaridis, I., Ginige, A.: Framework for collaborative web applications. LNCS, pp. 539–544. Springer, Heidelberg (2005)
3. Ahn, G.J., Mohan, B.: Secure information sharing using role-based delegation. In: Proceedings of the seventh ACM symposium on Access control models and technologies. tY - CONF., pp. 810–815 (2004);
4. Charles, J., Phillips, E., Ting, T., Demurjian, S.A.: Information sharing and security in dynamic coalitions. In: Proceedings of the seventh ACM symposium on Access control models and technologies (SACMAT 2002). SACMAT, pp. 87–96 (2002)
5. Seidmann, A., Sundararajan, A.: Building and sustaining interorganizational information sharing relationships: the competitive impact of interfacing supply chain operations with marketing strategy. In: Proceedings of the eighteenth international conference on Information systems (ICIS 1997), pp. 205–222. Association for Information Systems, Atlanta (1997)
6. Edwards, W.K.: A framework for information sharing in collaborative applications. In: Proceedings of ACM CHI 1994 Conference on Human Factors in Computing Systems, ser. INTERACTIVE POSTERS. Human Factors in Computing Systems, vol. 2, pp. 89–90 (1994)

7. Marmaridis, I., Ginige, A.: Sharing information on the web using bitlets. In: Proceedings of the 6th international conference on Web engineering, pp. 185–192. ACM Press, New York (2006)
8. Liang, X., Ginige, A.: Smart business object - a new approach to model business objects for web applications. In: International Conference on Software and Data Technologies (ICSOFT 2006), vol. 2, pp. 30–39 (2006)
9. Lee, M.: Collaborating to Win - Creating an Effective Virtual Organsiation, Taipei, Taiwan, March 26-27 (2004)
10. Ginige, A., Murugesan, S., Kazanis, P.: A road map for successfully transforming smes into e-businesses. Cutter IT Journal 14 (2001)
11. Tripathi, A., Ahmed, T., Kulkarni, D., Kumar, R., Kashiramka, K.: Context-based secure resource access in pervasive computing environments. In: Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops. tY - CONF., pp. 159–163 (2004)
12. Park, J.S., Sandhu, R., Ahn, G.-J.: Role-based access control on the web. ACM Trans. Inf. Syst. Secur. 4(1), 37–71 (2001)
13. Lieberman, H., Paterno, F., Klann, M., Wulf, V.: End User Development, ser. Human-Computer Interaction Series, vol. 9, ch.1. Springer, Heidelberg (2006)
14. Ginige, A.: New Paradigm for Developing Evolutionary Software to Support E-Business. In: Handbook of Software Engineering and Knowledge Engineering, vol. 2, pp. 711–725. World Scientific, Singapore (2002)
15. Marmaridis, I., Ginige, J.A., Ginige, A., Arunatilaka, S.: Architecture for evolving and maintainable web information systems. In: Proceedings of the international resource management association conference, IRMA 2004 (2004)