

Towards End User Development of Web Applications for SMEs: A Component Based Approach

Jeewani A. Ginige, Buddhima De Silva, and Athula Ginige

University of Western Sydney, Locked Bag 1797, Penrith South DC, 1797, NSW, Australia
{achandra,bdesilva}@cit.uws.edu.au, a.ginige@uws.edu.au

Abstract. ‘Garbage in Garbage out!’ This is very true when user requirements are not addressed accurately in Web (or any) Application development. Adding fuel to this fire is the ever-changing business requirements that force these web (or any) applications to change and evolve. In order to deliver web applications that meet user requirements, within budget and time constraints, it is important to use appropriate methodologies, tools and techniques suitable for a dynamic environment. Experts in Software Development have been researching and practicing many approaches to overcome these issues. However, after a decade into Web application development, only a few of these approaches are effectively used in this domain. This paper discusses use of two such approaches: (a) use of Components and (b) End User development that can be effectively used in combination for Web Application development. In our work in the Small to Medium Enterprise (SME) sector we have developed a set of Components that allow End Users to assemble, deploy and run Web Applications. We have incorporated these Components into a framework that facilitates the deployment of these Components developed by developers.

1 Introduction

With advancements in Web in past decade, businesses have been using Web technologies to make a global presence and/or to enhance internal business processes. Web’s ubiquity, simplicity and cost effectiveness have made it a suitable tool especially for Small to Medium Enterprises (SMEs) to be competitive in a global business environment. AeIMS (Advance enterprise Information Management Systems) [1] research group at University of Western Sydney has been working with SMEs in Western Sydney region to investigate how Information and Communication Technologies (ICT) can be used to enhance their business processes to get competitive advantages in global economy[2]. In this work we have identified various issues that SMEs face when trying to implement web based systems to enhance their business processes. These issues vary from not being able to get web applications developed to meet needs of the business in a timely manner to development projects running over budget.

Researches and practitioners in Web domain have been trying to adapt solutions that were initially developed for software engineering [3]. In our research we find that using one approach, method, etc. in isolation does not address all the issues that we have come across in Web Application development in the context of SMEs. Ginige, A. [2] suggests that computers should be viewed as medium to capture knowledge instead of creating specific end products as a way of meeting needs of SMEs in relation to web applications to support their business processes. Fischer, G. also supports this view of End User development as future of software development [2, 4, 5]. IT

consultants also recommend, “do it your-self”, DIY, approach for SMEs [6]. Based on above thinking in this paper we present a solution that enables End Users to develop applications using Components that are developed by web developers.

We have identified two types of basic Components; they are (a) Tools that allow End Users to create and assemble applications and (b) Engines that could be used to run these applications. The concept of Components presents a need for a framework that can support execution of them. Therefore we have extended a web application development and deployment framework, Component Based E-application Deployment Shell, (CBEADS©) which was developed by AeIMS Research group. CBEADS© framework and generic Components allow End Users to assemble, deploy and run applications by themselves. Further more new Components could be added to this extensible framework by developers. End Users can reuse the Components to develop different other applications.

Major findings that we present in this paper are; a discussion of issues in relation to Web Application development for SMEs, combination of Component based approach with End User application assembly and deployment to overcome the issues that were identified, a study into finding requirements of aforesaid Components and extension of CBEADS© framework to incorporate Components (Tools and Engines).

2 Solving the Problem at Conceptual Level

In this section we analyse different issues in web application development in SME context followed by an investigation of web application domain to find how other researchers and practitioners have tried to solve similar issues. Then we bring forward a concept of incremental development environment where developers and End Users can work together (not necessarily in the same location) to develop applications to overcome the issues that were identified.

2.1 Web Application Development Issues in SMEs

As a part of our work in SME sector in Western Sydney region of Australia, we have been assisting organisations to eTransform their business activities [7]. In the past three (3) years we have been closely working with fourteen (14) such SMEs. With these companies we have developed seventeen (17) web applications in total, ranging from static public web sites to intranet applications to support their eTransformation process. This work with SMEs provides a test-bed for the three strands of research namely eTransformation, eApplication Development and Emerging Technologies carried out by AeIMS [1] research group. Based on the analysis done on web applications developed so far and also with our previous experiences in software application development we identified the following issues that are related to web application development in the context of SMEs;

- Developers failing to fully understand user requirements.
- Users failing to accurately specify changing requirements in evolving business environment and developers failing to incorporate them successfully into already developed applications.
- Need for incremental addition of new web applications to keep pace with eTransformation stages.

- Keeping up-to-date information in web applications that are visible to a wider audience.
- Web Application development projects failing to deliver on time and within budget.

To address some of the above issues, in past decade experts in web domain have been researching and adapting some concepts used in software engineering into web applications development. In particular researchers such as Li [8], Petkovic et al [9], Gellersen et al [10], Zhao [11] have used Component based approaches for web application development. Also researchers such as Rode et al [12] and Morishima et al [13] have explored suitability of End User computing for web application development. However, we find that use of one approach, methodology, tool or technique in isolation cannot address all the issues of web application development mentioned above.

2.2 Proposed Solution: Combining End User Development with Component Based Development Approach

We studied approaches SMEs have been using to develop different web applications. Applications varied from a simple static web site to an automated internal business process such as a leave approval process, enterprise resource planning etc. Mainly SMEs use one of two methods (Fig 1) in order to develop these applications. First method (see (a) in Fig 1) is to get an analyst to specify application requirements, get designers to design the system and developers to implement it. In reality above tasks are done either by an individual or a group of people, depending on the complexity of applications. Other extreme (see (c) in Fig 1) is to use a tool that allows them to attempt to do this task by themselves. The WYSIWYG type web development tools such as FrontPage and Dreamweaver make it possible for End Users to develop simple web applications without having to do any coding or relying on a developer.

Both above approaches ((a) and (c) in Fig 1) have their own problems. Success of resolving web application development issues in the first method depends on many factors such as how well the users communicate requirements to analyst, skills of design and development team, appropriateness of methodologies used and flexibility of design to incorporate future needs. These factors are beyond the control of SMEs and do not guarantee to solve web development issues outlined in section 2.1 above. On the other hand second option ((c) in Fig 1) makes it possible for users to develop these applications themselves. Hence we can assume issues of communicating user requirements to analysts are eliminated. However, most tools that are available for End User web application development are either not capable of developing complex applications with both front end and back end components or way too complex to be used by nonprogrammers (End Users). Copeland et al's analysis on existing web development tools verifies this observation [14].

Therefore to strike a balance between above two extremes ((a) and (c) in Fig 1) we suggest a hybrid approach. In this approach developers create Components (Tools and Engines), which End Users can use to create and deploy web applications. This hybrid approach also supports incremental development. End users can develop applications when required and also maintain them. Developers can keep adding new Tools and Engines when existing Components cannot meet End User needs.

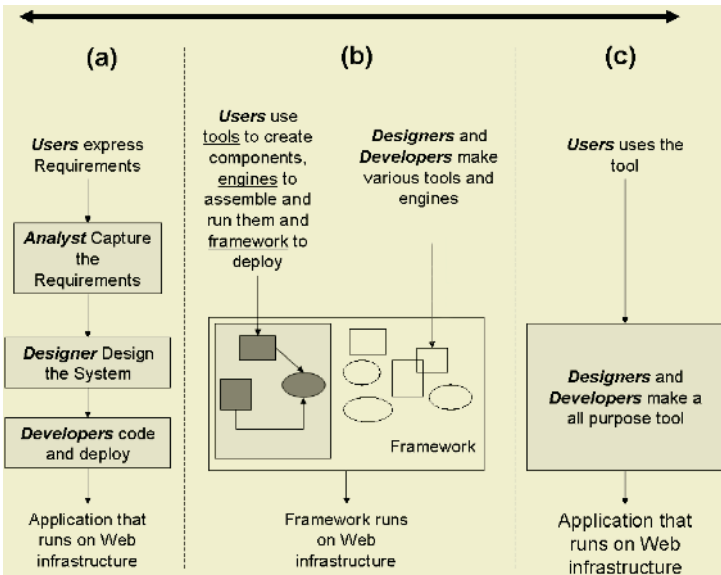


Fig. 1. Possible Approaches for Web Application Development

We believe the use of End User development concept and Component based development approach in combination in an incremental development environment address the issues in web application development in the context of SMEs. Also the reusability aspect of Components (Tools and Engines) makes it possible to rapidly develop web applications within budget. Since these Tools and Engines are agile and supports evolution, End Users themselves could handle change management of applications. Further the incremental development environment allows SMEs to meet new web application needs required in eTransformation [7].

3 Design of Tools and Run Time Engines

In this section first we discuss End User perspective of technical concerns in web application development. This is followed by a discussion on how to encapsulate critical application development know-how instead of application domain knowledge into Components so that End Users can use their application domain knowledge and develop their own web applications. In this approach these Components will be re-used across different web applications as we provide generic Components. Further we present implementation of these Components in CBEADS[®] framework.

3.1 End User Perspective of Technical Concerns of Web Application

In Rode et al's [15] empirical research into 'End Users mental model' they attempted to find functionalities that web application development tools should provide to handle concepts that are critical to web application development. We have extended and categorized their list of critical concerns in web application development (see Table 1 left most column). The concerns critical to web application development are mainly

three fold. They are issues in (i) front end development, (ii) back end application development to support the front end and (iii) data layer that supports both backend and front end functionalities. Also in Rode et al's research they have accurately identified that tools provided for End Users to develop web applications should encapsulate issues critical to web domain. Further we argue that it is important to abstract user requirements and develop a set of Components that does not depend on user application domain. This method allows those Components to be used across many application domains to develop applications.

Ginige and Murugesan have researched into categories of web applications and according to their classification they had identified six types of web application categories [16]. We have reorganized this categorization and added few more to it such as web-based application development tools and search tools to bring it up to date.

In Table 1, we have attempted to bring together Rode et al's [15] critical issues of web applications development (left most column of Table 1) with Ginige and Murugesan's [16] types of web applications (top most row of Table 1). Aim of this exercise was to find degree of importance of different concerns with various types of web applications (see Table 1). Then we asked eight (8) experienced web application developers to fill cells in the middle with a rating to identify to what degree each of the critical issues are applicable to different types of web applications. After correlating their input we produced Table 1.

Use of the information gathered in Table 1 is to find different critical concerns that should be supported by a framework and by Components (Tools and Engines). For example a framework should support concerns that are common to all web applications while other issues specific to individual application types need to be addressed by Components. Using this approach we can hide complexities of web application development from End Users and provide a framework that is easy to learn and use. The requirements for Components are found by abstracting different applications, for example if the need for the application is to develop a 'leave application approval process' the Components should be abstracted to a level up (Fig 2). This leads to providing a Tool that can generate forms and attach rules that apply to leave processing and an Engine that can support runtime operation of the form based on the rules.

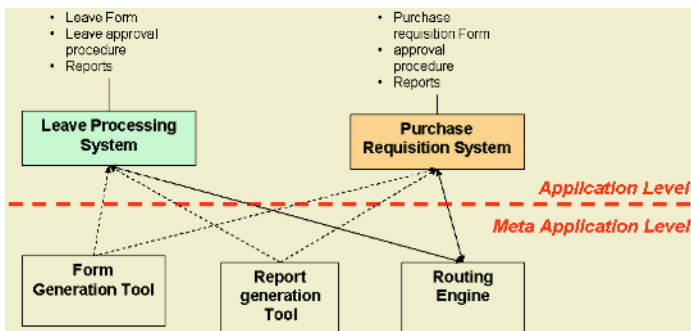


Fig. 2. Meta Level Abstraction of Component Parts of the Application: An example of re using Components (Form Generation Tool, Report Generation Tool and Routing Engine) developed for 'Leave Processing System' in a different application scenario (i.e. to develop a 'Purchase Requisition System')

3.2 Requirements of Components

In creating Components that are suitable for End User application development, developers have to take following requirements into consideration;

- Components should capture domain knowledge of web application development and hide complexities from End User.
- Should not capture application domain specific knowledge into Components. Rather those specific application needs should be abstracted and generic Components (Tools and Engines) should be created that can be used across many application domains (Fig 2).
- Components should be easy to use by End Users, yet they need to be complete so that it aid full capture of all the necessary ‘Components parts’ of the application such as front end pages, back end processing logic and database information.

3.3 Extensions to CBEADS[©]

Initial CBEADS[©] (Component Based E Application Deployment/Development Shell) [2], [17] was created in 1998 with basic functionality of user authentication, role based access control, ability to add more functions and assigning these newly created functions to users. This ability of adding more functions to CBEADS[©] made it a candidate for it to be the framework that can support incremental development concept. With the extension of CBEADS[©] with Components that are suitable for End User application development, now it has got four types of modules, as depicted in Fig 3; (a) tools, (b) engines, (c) applications and (d) the shell. Tools and Engines are the Components implemented by developers that can be used by End Users to assemble, deploy and run applications. Applications could be both End User developed or developer implemented applications that are deployed on CBEADS[©]. However, depending on the need applications could be developed that does not need to run on CBEADS[©] framework. For example a static web site generated using ‘content management system’ tools could be deployed with an ISP. The shell contains modules that are common to Tools, Engines and applications to operate smoothly. CBEADS[©] environment as a whole depend on basic web infrastructure such as operating systems, web servers, etc. for its operation.

When adding new Components to CBEADS[©] developers need to decide whether it need to be a tool, engine or a module in CBEADS[©] framework and register them accordingly. Tools are the Components that are needed for creating and assembling ‘Component parts’ of an application, for example front end pages, business rules, back end databases, etc. Engines are the Components that support runtime operation of the application. If all Tools, Engines or applications need a module it is implemented as a module in the core shell. For example functionalities such as user management, access control, session management, etc. are needed by Tools, Engines and applications. Hence they are in CBEADS[©] core shell.

4 An Application Scenario

Using Tools and Engines on the extended CBEADS[©] framework we have enabled SMEs to develop and deploy web applications in an incremental manner. Here we

Table 1. Mapping of Critical Concerns of Web Application Development into different types Web Applications

Type of Web App. Critical Concerns in Web App. Dev.	Informational	Search, directory and look-up tools	E-commerce sites and Web portals	Entertainment, interaction and messaging	Workflow or collaborative sites	Service oriented or high end web development tools
Site Structuring and navigation handling	medium-high	medium	high	high	high	high
Page Templates and Form creation	medium	medium	medium - high	medium-high	high	medium-high
Validating inputs	Low-medium	low-medium	Medium-high	Low-medium	high	high
Binding Form inputs with back end databases/tables	low-medium	Low-medium	medium-high	Medium-high	high	high
User authentication and access control	Low-medium	low	Medium-high	Low-medium	high	high
Session Management	Low-medium	low	Medium-high	Low-medium	Medium-high	Medium-high
Representation of business logic (rules) in applications	Low	Low	Medium-high	low	high	Medium
Querying a database and getting conditional outputs	medium-high	high	Medium-high	Medium-high	Medium-high	high
Creation of back end databases and other concerns in DB creation (schema, normalization, indexing, etc.)	low-medium	high	Medium-high	Medium-high	Medium-high	Medium-high
Security of data in storage and transmission	Medium-high	Low-medium	high	low	Medium-high	medium
Handling of multimedia (streaming, etc.)	Low-medium	Low-medium	Low-medium	high	Low-medium	medium

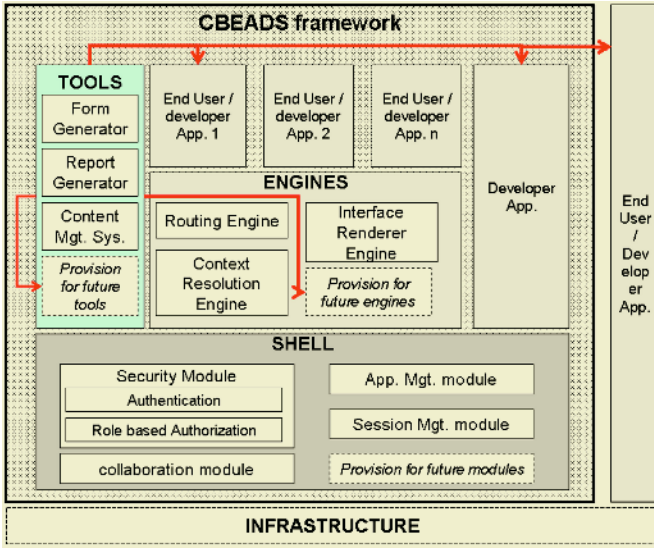


Fig. 3. CBEADS[®] framework; Tools, Engines, Applications and Shell

Table 2. Tools in CBEADS[®]

Tools	Critical concept covered	Description
Report Generator	Binding form input with backend database and tables	Use to design reports by defining data and calculations or processing want to apply on data retrieved.
Form Generator	Form Creation & validate inputs	Use to design web forms.
Template Generator	Template creation	Use to design the page templates.

Table 3. Engines in CBEADS[®]

Engine	Critical concept covered	Description
Routing engine	Representation of business logic (rules) in applications	Use to route a form from one user to another based on business rules.
Context Resolution engine	Querying a database & getting conditional outputs.	Use to resolve the context of artifacts. E.g.: immediate supervisor of given person.
Interface Rendering Engine	Binding form Inputs with backend database tables & validating inputs	Use to render the GUIs with form, report, etc.

elaborate how End Users could use CBEADS[®], Form Generator, Report Generator and Routing Engine to develop a leave application. First we have to create a new application and name it as leave application. Through CBEADS[®] framework we can also assign users or user groups for the applications. Then we have to create a leave form with different fields such as no of days, from to, and leave type, etc. using Form Generator tool and attach rules with the leave form. For example Rules can be routing sequence – from whom to whom the application and result has to be routed and when and what has to be saved to a database etc as shown in Fig 4. Next the leave applica-

tion should be registered with the routine engine, the run time Component responsible for routine forms. After that report formats and data to be displayed need to be specified. For example the data can be the statistics of leave applications processed with in a given period and the constraint is time period. Finally the reports are registered with the report generator to create the reports dynamically.

This section illustrates how the combination of component approach and end user development address the issues in web application development based on the above Application Scenario.

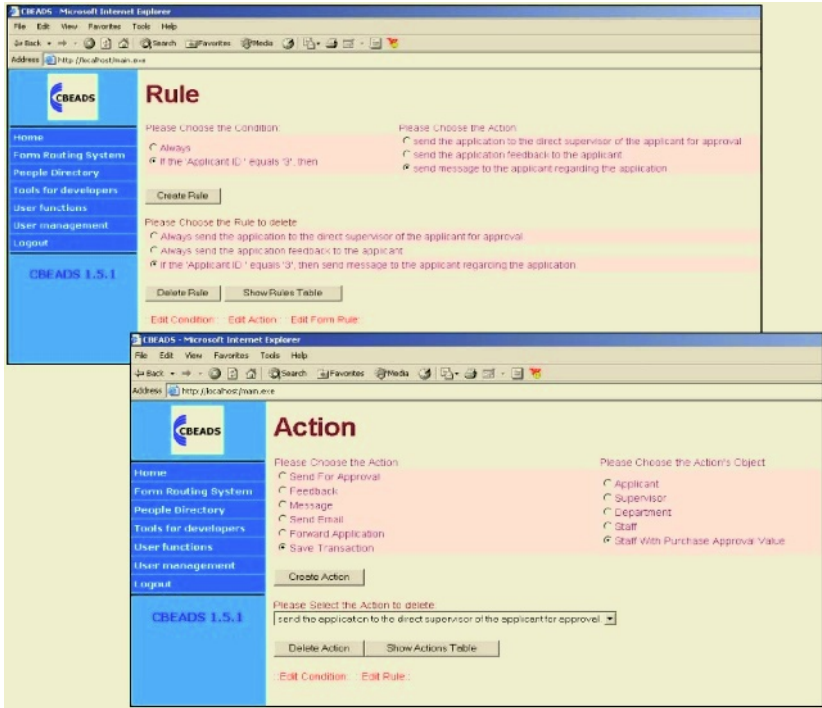


Fig. 4. Setting up Rules and Actions in Form Generator

Possible changes that could be requested after implementation for this leave application would be;

- Changing the initial leave form to add or remove fields
- Variations that could occur due to people (role) changes in the organization structure
- Changing the routing and business rules for example “Form 2005 onwards every leave application that is more than 15 days needs to be approved by the Managing Director”

Since the leave application implementation was abstracted to a level up to allow the changes related to the form itself (fields), its associated business rules and organisational structure to change individually without affecting one another it is easy for the above changes to be implemented to the leave application process.

In the case where there is a need for another leave application form for a sub company (with different organizational structure having changed roles and dissimilar routing and business rules) or a new form (such as purchase requisition form) for the same company, it is easy to implement that system using the same tools that was used for leave application process. Hence it reduces the cost and the time involved in future similar development activities.

As rules can be added through GUI as an expression this allows end users who has got a better understanding of their own application rather than the developers to participate in development. This understanding helps them to improve the applications for their need and to specify requirements for new applications correctly.

The CBEADS[©] framework supports the incremental addition of applications. We can add, delete, update applications while the framework is in use.

5 Conclusion and Future Work

This paper discusses a Component based End User development approach which is suitable for SMEs. By providing End Users with a set of tools to develop the web applications themselves we can solve the problems with requirement analysis and also reduce the cost and time to develop the applications. However, the mindset of End Users should change for this approach to be successful, as End Users have to actively participate in design and development.

We are currently refining the Components to reduce the effort required by End User when developing applications and providing a feed back mechanism to guide End Users to minimize faults that can happen in web application development. Also another area that needs further investigation is handling of exceptions for an instance of a form (business process) that is in operation. At the same time we are researching on success of the approach with group of SMEs volunteered to develop their applications themselves.

Acknowledgements

The authors wish to thank Dr. Uma Srinivasan of Phi Systems for her invaluable comments and support throughout the writing of this paper.

References

1. AeIMS Research Group Web Site – UWS. last accessed on April, 2005 at <http://aeims.uws.edu.au/>, Published by Advance enterprise Information Management Systems – University of Western Sydney: Sydney.
2. Ginige, A. Re Engineering Software Development Process for eBusiness Application Development. in Software Engineering and Knowledge Engineering Conference – SEKE02. 2002. San Francisco Bay, USA.
3. Botting, R.J., A Glossary of Software Development methods processes and techniques. 2003, California State University: San Bernardino.
4. Fischer, G. and E. Giaccardi, A framework for the future of end user development, in End User Development: Empowering People to flexibly Employ Advanced Information and Communication Technology, V. Wulf, Editor. 2004, Kluwer Academic Publishers.

5. Fischer, G., et al., Meta Design: A Manifesto for End-User Development, in Communications of the ACM. 2004. p. 33-37.
6. Hason, H., "Cut Price" IT Consultants for SMEs, in Asian Small and Medium Enterprises in the information Age: Asian Perspective. 2001.
7. Arunatileka, S. and A. Ginige, Applying Seven E's in eTransformation to Manufacturing Sector, in eChallenges. 2004.
8. Li, Q., J. Chen, and P. Chen, Developing an E-Commerce Application by Using Content Component Model, in IEEE. 2000.
9. Petkovic, I.M., Component Development of the Client Side of the Web Applications, in TELSIKS 2003. 2003: Serbia and Montenegro.
10. Gellersen, H.W., et al., Patterns and Components: Capturing the Lasting amidst the Changes, in Active Web Conference. 1999: UK.
11. Zhao, W. and J. Chen, CoOWA: A Component Oriented Web Application Model, in 31st International Conference on Technology of Object-Oriented Language and Systems. 1999, IEEE: Nanjing, China.
12. Rode, J. and M.B. Rosson, Programing at Runtime: Requirements and Paradigms for Non-programmer Web Application Development, in Human Centric Computing Languages and Environments-2003. 2003, IEEE Symposium: Auckland, New Zealand.
13. Morishima, A., et al., Enabling End Users to Construct Data-intensive Web sites from XML repositories: An Example based approach, in 27th VLDB Conference. 2001: Roma, Italy.
14. Copeland, D.R., et al., Which Web Development Tool is Right for You?, in IT Pro. 2000. p. 20-27.
15. Rode, J., M.B. Rosson, and M.A. Perez-Quinones, End-Users' Mental Models of Concepts Critical to Web Application Development, in 2004 IEEE Symposium on Visual Languages and Human Centric Computing (VLHCC'04). 2004, IEEE Computer Society: Roma, Italy.
16. Ginige, A. and S. Murugesan, Web Engineering: An Introduction, in IEEE Multimedia. 2001. p. 14-18.
17. Ginige, A., New Paradigm for Developing Evolutionary Software to Support E-Business, in Handbook of Software Engineering and Knowledge Engineering, S.K. Chang, Editor. 2002, World Scientific Publishing Co. Pte. Ltd.: Singapore. p. pp. 711-725.