Integrating Advanced Collaborative Capabilities into Web-Based Word Processors

Haifeng Shen, Steven Xia, and Chengzheng Sun

School of Computer Engineering
Nanyang Technological University
50 Nanyang Avenue, Singapore 639798
{ashfshen,stevenxia,czsun}@ntu.edu.sg

Abstract. With the development of new web technologies, web-based collaborative applications, exemplified by office applications, are emerging to take advantage of web's attractive features. We propose to plug a collaborative engine into web-based office applications so that advanced collaboration capabilities can be seamlessly integrated without compromising or modifying their conventional capabilities. This engine lies on application-independent data addressing and operation models to be reusable for a wide range of diverse applications without being modified. In this paper, we present a data addressing model for web-based word processors, which complies with the model used by the engine and shall lay a good foundation for investigating data addressing models for other web-based applications.

1 Introduction

Web-based (or essentially web-browser-based) applications have been an alternative to desktop applications since the emergence of WWW and are becoming more and more attractive owing to its advantages like no installation, easy-to-use, ubiquitous accessibility, platform agnosticism, low risk of data loss, and more important, sharing and collaboration. Representative niche web-based applications include Internet applications such as web-based email, bookmarks, discussion boards, blogs, wikis, and search engines; e-Business applications such as e-Bay and Amazon; and various MIS (Management Information System) and ERP (Enterprise Resource Planning) applications. These applications use standard web and database technologies to provide a lightweight solution to average end-users who possess limited or fair set of skills of using computers.

Recent development of new web technologies exemplified by Ajax (Asynchronous Javascript And XML) and Web 2.0 is making it possible to port some widely-used desktop applications onto the web. These web-based applications have similar user interfaces, features, and functions as those offered by their desktop counterparts. Although their functions are relatively limited and not comparable with their desktop counterparts at the moment, they do provide essential functions that satisfy the majority of average end-users, and more importantly, these web-based applications have so many attractive advantages that do not exist in their desktop counterparts.

Y. Luo (Ed.): CDVE 2007, LNCS 4674, pp. 1–8, 2007.

Such applications are represented by various web-based office applications such as web-based word processors (their most influential desktop counterpart is *Microsoft Word*): *Google Docs* [1], *FCKeditor* [2], *Zoho Writer*, and *ajaxWrite*; spreadsheet authoring tools (their most influential desktop counterpart is *Microsoft Excel*): *Google Spreadsheet*, *Zoho Sheet*, and *ajaxXLS*; and slides authoring and presentation tools (their most influential desktop counterpart is *Microsoft PowerPoint*): *Zoho Show* and *ajaxPresents*.

Sharing and collaboration are two characteristic features of these web-based office applications. All of them support sharing of documents on the web in that a document can be accessed with nothing but a web browser by anyone authorized, at anytime, from anywhere. Some (e.g., *Google Docs* & *Spreadsheet*, *Zoho Writer* & *Sheet*) come with built-in collaboration functionality that allows multiple users to view and edit shared documents in real time. However, the collaboration functionality in these applications is limited in that it is either based on a sequential interaction paradigm, where only one user can modify the shared document at any instance of time, or based on a Copy-Modify-Merge paradigm (supported by an *html diff* and an *html merge* algorithm) , where concurrent conflicting changes will be aborted.

We propose to plug a collaborative engine into web-based office applications so that advanced collaboration capabilities, such as fast local response, concurrent work and unconstrained interaction, relaxed WYSIWIS (What You See Is What I See) [3], collaborative undo, and detailed workspace awareness, can be seamlessly integrated without compromising or modifying their conventional capabilities. This engine uses application-independent data addressing and operation models to underpin advanced collaborative techniques so that it can be plugged into a wide range of diverse applications without being modified.

As different applications use their own data addressing and operation models to reference and manipulate internal data objects, an application-dependent adapter middleware is required for the engine to be plugged into an application, which essentially bridges the gap between the data and operation models used by the application and engine. Therefore, we choose to study the data addressing and operation models for web-based word processors in order to explore a data addressing model and an operation model that rigidly comply with the models used by the collaborative engine. In this paper, we present a data addressing model for the adapter of web-based word processors, which complies with the model used by the engine and shall lay a good foundation for investigating adapters for other web-based applications (e.g., web-based spreadsheet authoring tools, web-based slides authoring and presentation tools). This model has been implemented in the adapter for an open source web-based word processor, which, when the collaborative engine is plugged in via the adapter, supports advanced collaboration capabilities.

The rest of the paper is organized as follows. The next section describes the cornerstone of the collaborative engine. After that, we present a data addressing model for web-based word processors. Finally, the paper is concluded with a summary of major contributions and future work.