Integration and Communication of Process Support Tools in an Online Virtual Learning Environment

L. Raeside

B. Busschots

Shen Fei

J. G. Keating

Follow this and additional works at: https://arrow.tudublin.ie/itbj

Part of the Computer Engineering Commons

Integration and Communication of Process Support Tools in an online Virtual Learning Environment

L. Raeside
Department of Informatics and Enginneering,
Institute of Technology Blanchardstown,
Blanchardstown Road North, Blanchardstown,
Dublin 15, Ireland

B. Busschots, Shen Fei, J. G. Keating
Department of Computer Science,
National University of Ireland, Maynooth,
Maynooth, Co. Kildare, Ireland
May 7, 2006

Abstract

The Education through Virtual Experience (EVE) research group based at the National University of Ireland Maynooth have successfully developed an online Virtual Learning Environment (VLE) for Astronomy as part of the Virtual Telescopes in Education (VTIE) project. The VTIE VLE has been developed to provide online software support for the engagement of schoolchildren and their teachers in a scientific process. The development of an online VLE to support a scientific process has brought about complex interactions from the user perspective. This has lead to complex integration and communication challenges from the software implementation perspective. This paper contains a high level user-centered summary of the integration and communication challenges that have arisen during the VLE development and summarizes the mechanisms used to meet these challenges.

1. Introduction

The VTIE VLE, the virtual learning environment discussed in this paper, has already been used successfully to engage schoolchildren in a scientific process through Astronomy based experimentation [1]. The VLE has been constructed using a rapid prototyping approach combined with formative evaluations [2]. The VLE provides a software environment through which schoolchildren and their teachers can create project proposals, investigate and analyze data and images associated with that proposal and engage in team based collaborative writing to summarize all of the work carried out during the course of a project [3]. The completion of the VTIE scientific process culminates in the production of a project research paper. Following the completion of the formative evaluations of the VLE, the requirement for a set of fully integrated process support tools was identified. The aim of these tools is to fully support of all the above phases of the VTIE scientific process from project proposal to the generation of the project research paper. These tools are required to work online and be consistent in terms of look-and-feel, interoperability and usability within the VLE. The EVE research team have developed a number of tools to support the scientific process including the Collaborative Writing Environment (the CWE includes the Project Design and Management

Interfaces) [4], the Scrapbook Tool, the Imaging Tool [5] and the Graphing Tool. The CWE allows students to produce a collaboratively written document within the VTIE environment that amalgamates all of the experiences of a group of students working on a particular project into a research paper. The Scrapbook Tool provides the basis for sharing of information resources between students within the VLE. The Imaging Tool provides a mechanism for students to analyze images and produce data about the images being analyzed. Finally, the Graphing Tool ensures that student data can be shown graphically via plots and graphs, these graphs are added to the group research paper. This paper summarizes the software design constraints applied to the VLE in order to ensure consistent interoperability between the process support tools.

2. VTIE VLE Architecture

The VTIE VLE is a Java-based solution that has been constructed to operate within the Firefox web browser. The Firefox browser is part of the Mozilla project, is available online and may be downloaded and installed by participating schools at no cost [6]. There are a number of features that make the Firefox browser a suitable front-end for the VTIE VLE including the provision for rich customizability through software extensions, the provision of separate browser tabs which enhance the usability while navigating the VLE, and a high level of compatibility with the W3C standards [7]. In order to maximize the ease of integration of tools within VTIE, the VLE has been implemented using the Apache Struts framework [8]. The Apache Struts framework is based on the Model View Controller Model 2 (MVC2) design pattern and offers a well structured architecture through which new tools can be seamlessly introduced [9]. Each new software tool introduced into the VLE must follow the Struts framework specification, i.e., JSP is used for the view and the model is implemented using Java classes (namely Action classes). The Struts Controller is implemented as a servlet that maps events to particular classes using an XML based configuration file. The integration of new tools involves the editing of this configuration file and the addition of server classes to implement program logic [10]. Access to the VTIE database is carried out in a unified manner using Java utility classes designed at the architectural level. This ensures that each of the software tools can achieve database access without having to implement the low-level code statements to store and share information. The VTIE utility packages also provide unified access to various session-related information such as direct access to information relating to the current users and user groups. Communication between the various tools within the VLE is achieved using XML. The front-end communicates with the server via AJAX [11] and the server based components use the JDOM package [12] to produce and parse XML schema. The XML schema are used to represent data passed between tools and between the client and

server. Figure 1 shows a diagram of the VTIE VLE Architecture and the communications mechanisms between the client and server components.
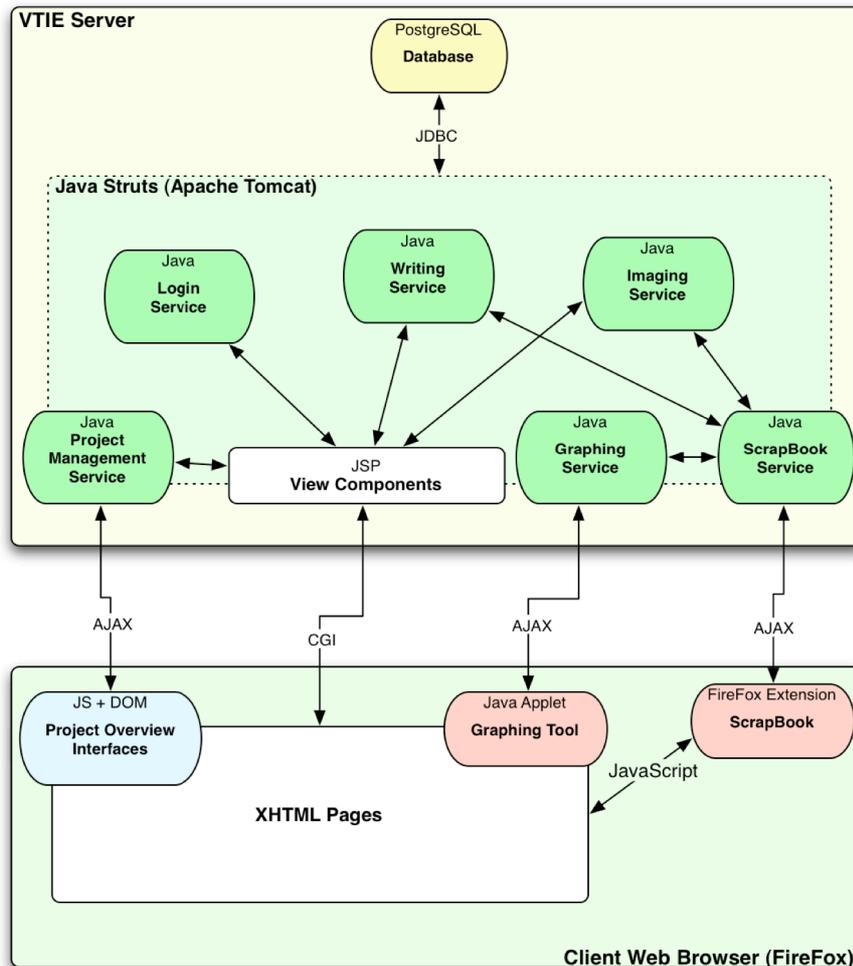

Figure 1: Architectural Diagram of the VTIE VLE

3. *Examples of Tool Integration and Communication in VTIE*

The requirement to support the VTIE scientific process has been the determining factor with respect to the communications between the various software tools. The communication links between the various tools have been formed directly from the user requirements and the educational goals of the VLE. This section provides examples of some of the communication that takes place between the process support tools within VTIE.

The Project Design and Management interface (a component within the CWE) is responsible for initializing projects, setting up assignments, assigning students into various groups and the assignment of members of each group to particular aspects of each project. The Project Design and Management interface is used primarily by the students' mentor, the mentor may be a teacher or any other person acting as the coordinator of student groups. The mentor can use this interface to assign particular sections of a document to particular members of each student group. The students then work in their respective teams while

following the VTIE scientific process described above. The UML Use Case diagram in Figure 2 represents the main functions that each student carries out while completing a project. The Write Paper Use Case shown in Figure 2 involves the use of the Scrapbook Tool to gather various resources and include them in the project research paper, it also involves the use of the Collaborative Writing interface within the CWE and may also require the use of the Graphing Tool if the student wishes to produce a graph of project data. The Analyze Images Use Case shown in Figure 2 involves the launching of the Imaging Tool. The student may have stored images relating to the project and may wish to carry out measurements on these images. The Browse Web Use Case involves the use of the Firefox browser to search other online resources for information relating to the project, this phase of the project uses the Scrapbook Tool to store URLs, text, and any other information relating to the project. The Scrapbook Tool is capable of persistently storing any online resource while students browse the Web. This information is stored automatically in the VTIE database and can be shared within the students' group and may also be viewed by the mentor.
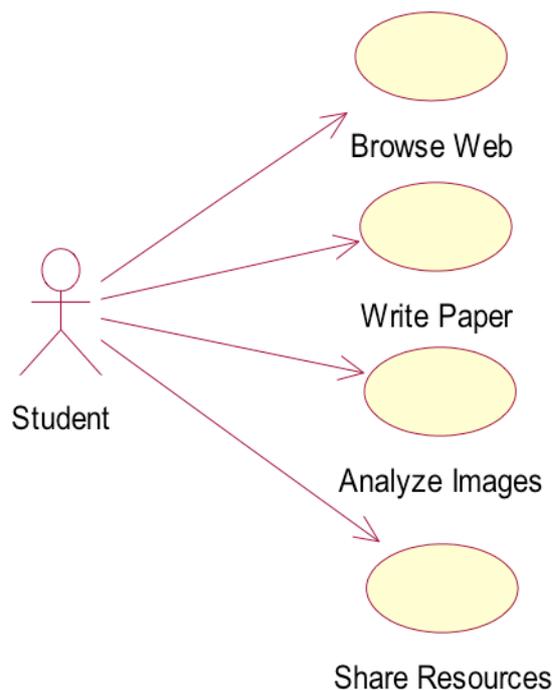

Figure 2: UML Use Case diagram of main activities for the student within the VTIE Process

The Scrapbook Tool is implemented as a Firefox extension. This tool operates within the browser window and can be used within the VTIE environment. However, this tool can also be used as a general purpose tool for the storage and retrieval of online resources within Firefox. The UML Use Case diagram in Figure 3 represents a student-centered view of the main functions performed through the Scrapbook Tool. The Scrapbook Tool integrates and

communicates with all of the other tools within the VTIE VLE and provides a universal linking mechanism to the project research paper. The Scrapbook Tool represents an example of a communication intermediary between other tools within the VTIE VLE. The Scrapbook Tool communicates with other tools in the VLE through direct interaction with the students and is constantly available in the left pane of the browser window. Students simply carry out click-and-drag operations on online resources from the main browser window into the Scrapbook Tool as required. The student resources are automatically saved to the database and are associated with either the individual student or the student's group. The Scrapbook Tool is also used as an intermediate communication link between the Imaging Tool and the Graphing Tool and the research paper created using the CWE. Images that have been modified within the Imaging Tool and graphs that have been generated within the Graphing Tool can be saved directly to the Scrapbook Tool as with all other online resources. The Add Scrap To Paper Use Case shown in Figure 3 allows students to add any of the resources collected into the project research paper including those images and plots generated within the VLE.
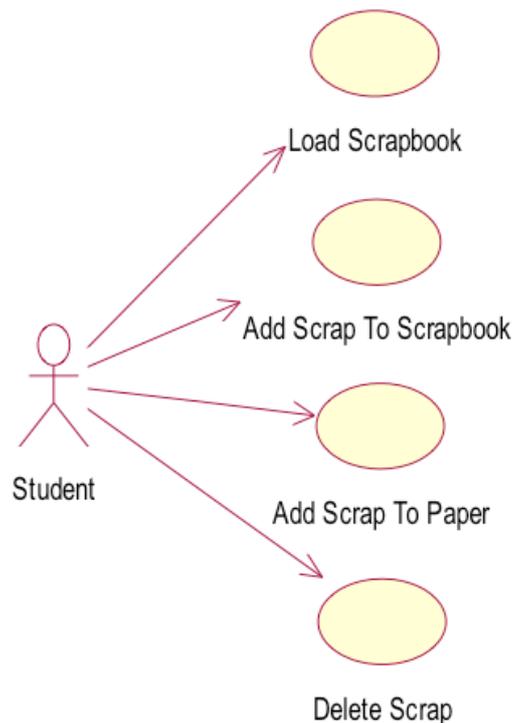

Figure 3: UML Use Case diagram of main functionality provided by the VTIE Scrapbook Tool

The analysis of images using the Imaging Tool and the Graphing Tool represents a direct tool-to-tool communication link within the VTIE VLE using the VTIE database and XML. The UML Use Case diagrams in Figures 4 and 5 represent a student-centered view of the functionality provided by the Imaging and Graphing Tools respectively. As part of the

analysis phase of the process the student must be able to gather data about images and then plot that data when required. Students may choose to gather data as one brief activity or they may choose to accumulate data over many days, weeks and perhaps even longer periods of time. In order to cater for the various time lapsed experiments that students may perform using VTIE it is important that data associated with images be stored persistently and that this data be associated with the student, the project and the image being analyzed. The Imaging Tool provides the functionality to store data associated with the student images in the database using an XML representation.

Figure 6 shows an example of the type of information stored in the database, the XML instance shown represents an angular measure carried out using the Imaging Tool (an angle measure requires the storage of three points in the image and the angle calculated between those points using the Imaging Tool). It is this XML database representation that forms the communication link between the Imaging Tool and Graphing Tool. When the student collects the necessary data from the image being analyzed this data is stored in XML format into the database, subsequently, the Graphing Tool may be launched by the student via the Imaging Tool or from elsewhere within the VTIE VLE. When the student chooses to launch the Graphing Tool directly from within the Imaging Tool an XML representation of all of the data saved for the current image is retrieved by the Graphing Tool from the database.

The Graphing Tool retrieves the XML representation of the image being analyzed and uses JDOM to extract the image data for display within the data interface component of the Graphing Tool. Finally, the options interface of the Graphing Tool allows the student to select and modify the data they wish to display so that the graph is of the desired format to be added to the project research paper. The Graphing Tool provides the functionality to store the current plot as an image representation that can be retrieved, stored, and shared via the VTIE Scrapbook Tool as described above.
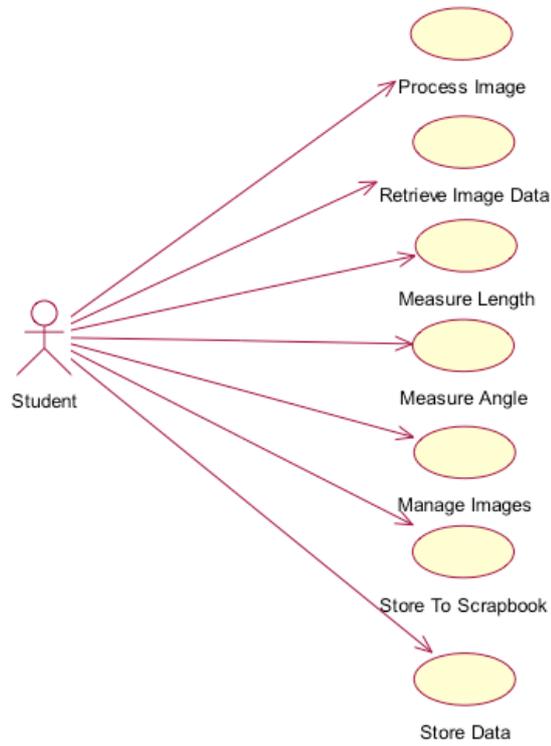
Figure 4: UML Use Case diagram of main functionality provided by the VTIE Imaging Tool

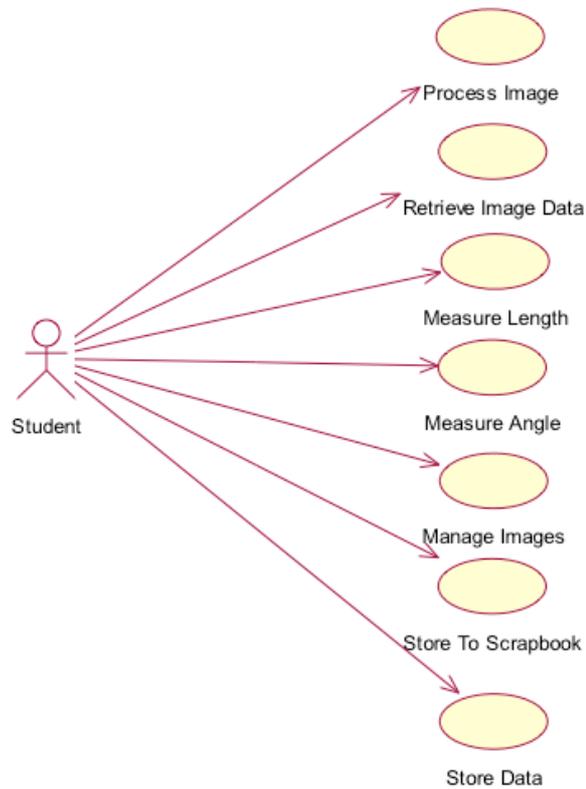**Figure 5: UML Use Case diagram of main functionality provided by the VTIE Graphing Tool**

```
<angleMeasure>
<innerAngle> 134 </angleMeasure>
<outerAngle> 226 </outerAngle>
<centerX> 200 </centerX>
<centerY> 198 </centerY>
<x1> 157 </x1>
<x2> 117 </x2>
<y1> 278 </y1>
<y2> 219 </y2>
</angleMeasure>
```

Figure 6: XML Representation of and Angle measurement using the VTIE Imaging Tool

4. Conclusions

The formative evaluations of the VTIE VLE have demonstrated that it is possible to successfully engage schoolchildren and their teachers in a scientific process using an online environment. The software required to support this scientific process has been continuously evolving in response to user testing. The complexity of the user interactions during the engagement within the scientific process has necessitated the implementation of complex integration and communication links between the various process support tools in the VLE. This paper has provided a high level user-centered model of the integration and communication necessary to support the various stages of the VTIE process from which more detailed models will emerge. The modeling of the integration and communication links between the tools within the VTIE VLE will provide the EVE research team with a valuable representation from which to base future development. Finally, the integration of the VTIE process support tools has served to illustrate the importance of applying sound software engineering practices to the development of a complex learning environment.

References

[1.] Hoban, S., desJardins, M., Farrell, N., Rathod, P., Sachs, J., Sansare, S., Yesha, Y., Keating, J. G., Busschots, B., Means, J., Clark, G., Mayo, L., & Smith, W. (2002). Virtual Telescopes in Education. *Journal of Digital Information*, 2(4), Article No. 104, 2002-05-23.

[2.] B. Busschots, L. Raeside, J. G. Keating (2005). Formative evaluations of a virtual learning environment for Astronomy education. *Accepted Journal Paper: Computers and Education*

[3.] Hoban, S., Keating, J. G., Sachs, J., Laughlin, D., & Yesha, Y. (2003). Science Investigation System for Telescopes in Education Research. *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, Track 1 - Volume 1, p35.1.

[4.] B. Busschots, L. Raeside, S. Waddington, J. G. Keating (2006). The VTIE collaborative writing environment. *Proceedings of the 2nd International WEBIST Conference on Web Information Systems and Technologies,* Volume 2, P. 221-228

[5.] L. Raeside, B. Busschots, E. O'Cinneide, S. Foy, J. G. Keating (2005). Empowering schoolchildren to do astronomical science with images. *Proceedings SPIE* Vol. 5827, P. 669-678

[6.] getfirefox.com (2005). The Firefox web browser, http://www.getfirefox.com. Available online. Accessed on the 26th of March 2006

[7.] w3.org (2006). World Wide Web Consortium, http://www.w3.org. Available online. Accessed on the 5th of May 2006

[8.] apache.org (2005). The Struts Framework, http://www.apache.org/struts. Available online. Accessed on the 26th March 2006

[9.] javasoft.com (2005). The Model View Controller Model 2 Design Pattern, http://www.javasoft.com. Accessed on the 26th of March 2006

[10.] S. Speilman (2003). The Struts Framework. *Morgan Kaufmann*

[11.] xulplanet.com (2006). Advanced Javascript and XML, http://www.xulplanet.com. Accessed on the 26th of March 2006

[12.] jdom.org (2006). The JDOM XML solution, http://www.jdom.org. Accessed on the 26th of March 2006