2D and 3D Visualizations in WikiDev2.0

Marios Fokaefs, Diego Serrano, Brendan Tansey and Eleni Stroulia
Department of Computing Science
University of Alberta
{fokaefs, serranos, btansey, stroulia}@ualberta.ca

***Abstract*—Several types of 3D software visualizations have been developed to communicate information about the products of a software project and, sometimes, the development process itself. These visualizations have been limited in the degree of interactivity they enabled (primarily panning and zooming) and in their accessibility (since in most cases they assumed a particular client platform). In this paper we discuss our 3D visualization of the data collected and extracted in our collaborative software-development platform WikiDev2.0, developed in the Open Wonderland virtual world. The visualization adopts a city metaphor, similar to earlier work, but advances the state of the art by providing a web-accessible distributed 3D environment where multiple users can explore the same project. In this paper we discuss this visualization, which we call WikiDev3D, and we report on our preliminary findings about its effectiveness against the more traditional visualization of the original WikiDev2.0 tool.**

I. Introduction

Software development is a fundamentally collaborative activity. Project activities are carried out by different team members, sometimes distributed in different geographical locations and working asynchronously.

Several empirical studies have shown that team software development poses socio-technical challenges including, most importantly, inefficient communication and ineffective administration, as well as the introduction of bugs in the code due to inconsistent updates by minor developers [1], [2]. Moreover, there is an abundance of tools to support individual life-cycle activities, and not always do team members rely on the same tool set; the tools that each individual developer uses may be based on his particular needs and expertise. Thus, the community recognizes the need for distributed (team) repository-based tool support, as opposed to the traditional (individual) workspace-centric tools.

The repository view poses two important technical challenges. First is the problem of the frequency and the granularity of the information updating. Clearly, in order for developers to be productive they need "fresh" data, which implies that the changes made to the software by each developers have to be pushed, in very fine-grained chunks, to the repository. Such frequent small global updates can impose a substantial overhead on the process, therefore inferring the desired granularity of updates is a matter of research and will most likely be a function of the application domain and the team size and process. The second challenge arises due to the large amount of data produced and accumulated during the project life-cycle, currently siloed in the repositories of the various tools used by the team developers. This data contains important and useful information about the development and the maintenance of the software; therefore, the data has to be integrated, linked, analyzed and presented in a way that will support the team-members' decision making instead of overwhelming them.

In this paper, we discuss the visualizations supported by WikiDev2.0 [3], [4], our MediaWiki-based repository-style platform for supporting collaborative software development. WikiDev2.0 offers a wiki-based environment, which is lightweight and easy to learn. Using a RESTful approach, it provides an integration mechanism for a variety of data sources originating from a variety of tools. The platform also offers a variety of analytics components in order to link the available data, irrespective of the tools from which they originate, and to extract useful information for the developers. To present this information, WikiDev2.0 uses (a) traditional visualizations, such as line charts or pie charts, and (b) a novel 3D visualization built in the Open Wonderland ($http://www.openwonderland.org$) virtual world. The 2D views rely on tabular views of relational data and graph-based views correlating data in two dimensions, such as changes in time, or changes by developer. The 3D visualization adopts a city metaphor where different types of software and communication artifacts are visualized as different types of buildings; user contribution to these artifacts is shown as colored stripes on the buildings; development metrics - such as number of changes - are shown as the height of the buildings; and relationships between artifacts are reflected as proximity within city blocks.

The rest of the paper is organized as follows: in Section II we place our work in the context of related research. In Section III we present the architecture and the different parts of WikiDev2.0. In Sections IV and V we provide a detailed description for the 2D and 3D visualizations in the WikiDev2.0 platform. In Section VI we discuss our early empirical findings about the comparative merits of the 2D and 3D visualizations. Finally, in Section VII we conclude with a summary of the lessons we have learned to date and our plans for research in the near future.

II. Related Work

To place our work in the context of the state-of-the-art in the field, in this section, we review work in 3D visualization, looking specifically at empirical studies.

sv3D [5] is a software visualization application framework based on the SeeSoft pixel representation that extends that original metaphor by rendering the visualization in a 3D space.

26th IEEE International Conference on Software Maintenance in Timişoara, Romania

Software World [6] uses a real-world metaphor, with the world, countries, districts, and buildings as visual metaphors for the various parts of Java code. Both visualization systems visualize only static properties of code and have not been empirically evaluated; the sv3D developers argued for its usefulness by appealing to the usefulness of SeeSoft-like tools [7].

The next two systems developed 3D visualizations for software concerns other than static code structure. Greevy et al. [8] focused on the visualization of the run-time software behavior, extending a "traditional" 2D view of the software structure with a third dimension in which they visually represent the dynamic behavior of the structural elements at run time, namely object instantiations and message sending. Wettel and Lanza [9] adopted a city metaphor for their visualization of the structural evolution of object-oriented software systems, at different levels of granularity, aiming at supporting the comprehension of "the causalities which led to the current state of a system." This work inspired *Codstruction* ($http://marketplace.eclipse.org/content/codstruction-3dsoftware-visualization-tool$) a JMonkey-based implementation. The intent is similar to ours, in that we also want to enable insights on "why and how the system evolved to be what it is today"; however, we started by focusing more on understanding the collaborative software-development process as opposed to the logical evolution of the software products. In fact, we are now working on the integration of the logical-evolution aspect, in both WikiDev2.0 and WikiDev3D, by integrating our own work on UMLDiff [10] in the environment.

Martinez et al. [11] recently proposed a very rich (and rather complex) metaphor for visualizing the development process, including milestone trend analysis, planning deviations tracking, evaluation of error control procedures, assessment of change management policies and product quality estimation.

While it appears evident that 3D visualizations can potentially provide a far richer experience than 2D ones, there has been precious little empirical work actually evaluating this intuitive hypothesis. Koike and Chu [12] empirically studied the VRCS, a 3D visualization of RCS history, as 2D trees stacked in the z-axis, denoting time (a metaphor also adopted later by Greevy). They found that through VRCS users could be more effective in using the version-control system than through its original interface.

III. WikiDev2.0 Architecture and Features

WikiDev2.0 is based on a 3-tier architecture. The resource layer includes information about tickets, mail messages, IRC meeting logs, wiki pages, UML models and source code. This information is imported - in a manner transparent to the team - from Bugzilla, a dedicated mailing list, an IRC bot recording the development team channels and the team's SVN repository. The WikiDev2.0 platform supports a set of RESTful APIs for updating and accessing these resources; in this manner its resource layer remains independent of any external database schema or specific tool. The middle layer offers several services to support the team collaboration. These services include access control, object-oriented design evolution analysis [10], [13], and clustering of the resource assets. Finally, the top tier is the web-accessible front end, where all of the information is presented as wiki pages and interactive visualizations.

WikiDev2.0 is built on Annoki [3], our extension of MediaWiki, which offers a set of basic services, including (1) namespaces (to enhance the security and privacy of the project data); (2) a calendar (for scheduling support); and (3) templates (to define the structure of special-purpose wiki pages for progress documentation for example).

In addition, WikiDev2.0 includes additional support for collaborative software development. The **object-oriented design-evolution analysis** of JDevAn [10], [13] is currently being integrated in WikiDev2.0, with the aim of supporting the team's understanding of their progress in terms of the changes they bring about to the logical model of their project. The **artifact-clustering** [3] service identifies groups of strongly related artifacts including tickets, messages, wiki pages and code artifacts, based on text analysis. It can help the users answer questions like *what triggered a particular change*, *who discussed a specific artifact and should potentially be consulted about changes to it*, and *how might a developer's work affect others*.

IV. WikiDev2.0 2D Visualizations

The WikiDev2.0 platform presents the data collectively in tables and then combines these tables and presents the data in visualizations. The set of 2D visualizations of the WikiDev2.0 platform features traditional line and pie charts to represent contribution and communication data, in addition to an extended and interactive UML Viewer. We implemented these visualizations using the Google Visualization API and Adobe Flex.

The contribution-analysis visualizations use special metrics to keep track of each user's contributions to the project, in terms of source code commits, ticket changes and wiki page edits. This information can help developers answer questions like *who works on what and how much?* The contribution visualizations include line charts (Figure 1) for the activity across time, pie charts for the user contribution as a percentage of the total activity and pie charts for user contribution and ownership of individual artifacts.

The communication visualization communicates the intensity and directionality of the communication exchanges among the team members, in terms of email messages exchanged between team members, tickets on which multiple team members worked, co-edits of wiki pages and files changed by multiple members. We believe that this information can help developers answer questions like *who talks to whom and how much?*. Moreover, we are also interested in examining this information to identify communication patterns within a team, such as subsets of team members who work closely together on specific subsets of the project artifacts.

Finally, the UMLViewer is a Flash-based interactive display of UML models associated with a project. This extension uses XML representations of UML diagrams that have been

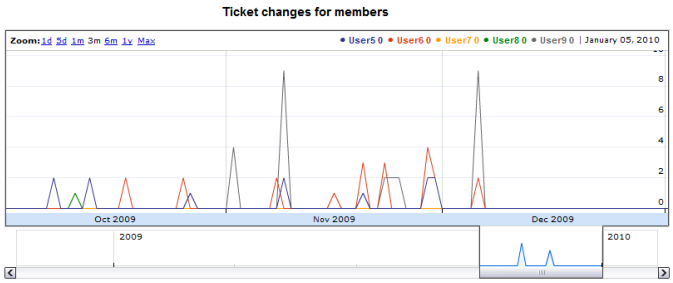

Fig. 1. Line chart which shows the development and communication activity across the project's life-cycle. The chart is zoomable.

converted to a JDEvAn [10], [13] database format, allowing for specific queries to be issued to display subsets of the diagram for further analysis. This viewer also uses the clustering data to display artifacts (such as mail messages) related to a specific UML object in order to provide a complete view of the development process for that object.

V. WikiDev2.0 3D Visualizations

WikiDev3D is a recent extension to WikiDev2.0 that provides a set of 3D visualizations of the information available in the system. We opted for a city metaphor, which allows us to present a great variety of information such as different types of artifacts, development metrics, user contribution and relationships. We also chose to implement the 3D visualization framework in Open Wonderland, an open-source java-based virtual world. In our own work, we have been exploring innovative uses of virtual worlds, as we believe that they have great potential as a software platform, based on their increasing popularity and adoption. Furthermore, a virtual world adds the notion of navigation through the data, which static 3D platforms lack.

We have divided our 3D environment in two cities: one for the communication artifacts of the project (messages, wikis and tickets) and another for the source artifacts (i.e. all the files in the SVN repository). The main reason for this division is that different types of artifacts have different relationships, although relationships between the two cities are also captured using the clustering information. While the two cities have certain differences, such as different buildings, different meanings for the height metrics and the relationships, they also have commonalities for consistency purposes. The method used to layout the buildings and organize them in city blocks is the same for both cities. Moreover, the roof and the first floors of each building in both cities are painted with stripes whose color corresponds to a user and the width to the contribution of each user as a percentage of the total number of edits that this artifact was subjected to. For the rest of this section the terms "building" and "artifact" are used interchangeably.

A. Communication City

The communication city contains all the tickets, messages and wiki pages created by the developers during the project life-cycle. The purpose of this city is to provide an overview of all the communication that took place during the development process. Furthermore, the visualization of the user contribution gives a quantitative index of who talked with whom and how much.

The height of each building is a function of a different property for each artifact type: for *tickets* priority; for *messages* the number of messages in the thread to which the message belongs; and for *wiki pages* the number of their edits.

The file city provides a visual representation of the SVN repository. Different types of files are represented by different buildings. In this city, the building height is calculated based on the number of changes that the corresponding file has suffered. The different types of files visualized as buildings include source, media, text, html and miscellaneous.

The buildings in both cities are grouped in blocks based on their similarity (textual for the communication buildings and co-change for the file buildings). For the layout of the buildings, we use mapping techniques, like MDS [14] and Sammon's mapping [15], which take distances as input and calculate the buildings' positions in the Euclidean space. We have also devised algorithms to ensure that the density information will be preserved while we distribute the buildings in the city blocks. More information about the layout of the cities can be found at $http://ssrg.cs.ualberta.ca/index.php?title=WikiDev3Dlayout$.

B. Project History Animation

WikiDev3D also supports animation. The user can see how many and what buildings were changed during the project life-cycle in order to get an overall impression of the project history. The user has the choice to automatically play the animation from the first week to the last one or jump from one week to the other. The buildings that correspond to artifacts that have changed between two subsequent weeks are painted red and their height changes accordingly. During the animation, buildings do not move around; instead, they are always placed in the location calculated based on their clustering for the last week.

C. The Open Wonderland Interface

The Wonderland platform supports several interactions enabling users to move inside the world, to inspect its contents and to talk with each other. In addition, for WikiDev3D, we also provided the users with a simple 2D interface, embedded in the world. Various kinds of information and controls are available to the user through a tabbed panel.

The **Properties** panel, when a user clicks on a building, shows general information about the visualized artifact, including

- the type and ID of the artifact, if it is a communication building;
- the height of the building and what it represents for the particular type;
- the users that have edited this artifact;
- the URL of the artifact;
- the date that it was created and last modified; and

- the user-name of the last editor.

Besides this information, the panel also contains a button linking to the wiki page of the artifact in the WikiDev2.0 system. The page is opened in a browser. From there the user can have access to the content of the artifact as well as additional information as the full history and contribution analysis. Moreover, this tab also has a button which prompts the Wonderland environment to visualize the clustering information, as this was calculated by the artifact-clustering module of WikiDev2.0.

The **Animation** tab controls the animation functionality of WikiDev3D. With the control buttons, the user can play, stop or pause the animation, go forward or backwards. The slider above the buttons is labeled with the weeks of the development. Therefore, the user can jump to a specific snapshot of the city. Finally, as the animation progresses, a label shows the total number of buildings changed from the previous week. This number, which refers to both cities, gives to the user a quantitative perception of the development activity for each week.

The **Visibility** tab is a legend of the colors corresponding to each developer. If the user checks the box next to a particular developer, only the buildings on which this developer worked will be shown and everything else will be hidden (Figure 2).

The **Search** panel enables the user to search for (a) a message by providing the subject of the thread and the id of the message; (b) a ticket by providing the ticket id; (c) a wiki page by providing the page id; and (d) file by providing the file-name. By selecting the "Hide others" option, users can hide all other buildings except the one they searched for. WikiDev3D paints a yellow antenna on the building to indicate the buildings that are part of the result of the search.

The **Cluster threshold** tab enables the user to control the artifact-clustering process. As we have already discussed in our earlier work [3], the artifact-clustering module requires a distance threshold. However, instead of requiring a user input value, we apply the clustering algorithm for threshold values ranging from 0.05 to 0.95. If the user selects the "Show clusters" button from the properties tab, the system paints yellow antennas on top of the buildings contained in the cluster. The brighter the yellow is, the lower the threshold value is, thus indicating a tighter connection. From the cluster threshold tab, the user can filter the distance threshold and hide buildings that are no longer in the selection. The user can uncheck a value if all the values that are greater than this are unchecked. This is because as the distance threshold increases more artifacts are added in the clusters. Thus, if an artifact is in the cluster for one value it is guaranteed that it will remain in the cluster for higher values. Moreover, in this tab the user can see the terms around which the artifacts where clustered, giving an indication of the topic or topics discussed by these artifacts.

VI. Evaluation

We conducted the pilot of a within-groups experiment to comparatively assess the usability of the two alternative

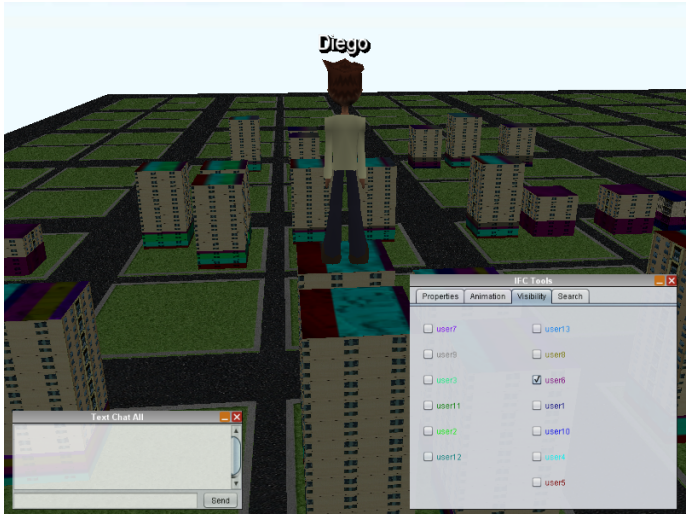

Fig. 2. View of the communication city and the visibility panel.

visualizations and their effectiveness in conveying useful information about the software project. We asked 4 individuals (students in Computer Science with no prior experience with the WikiDev2.0 system or the data visualized) to use the WikiDev2.0 system to answer some questions and compare the 2D and 3D visualization techniques.

The data that our subjects inspected was collected from a project that a different set of students developed in a Software-Engineering course in the past. This team had also used (an earlier version of) the WikiDev2.0 system to collaborate. The development of their project spanned for a period of 16 weeks. The data collected includes tickets, wiki pages, mail messages and SVN data.

We asked our subjects questions about (a) the communication and collaboration between developers (e.g who talked with user X or who collaborated with user X on file Z); (b) developer contribution (e.g who contributed the most to a file or to the project as a whole); and (c) development tasks and history of development (e.g. what was the most changed file or what was the file with the most editors). For each question, the subjects had to examine the data through the visualization corresponding to their condition, and, in addition, they had to say how easily they found the answer to this question in a scale from 1 to 4.

To evaluate the visualization effectiveness we identified the (set of) correct answers for each question. For some of the questions (e.g. give some of the oldest classes of the project) there were several equally valid answers; in these cases we considered the subject's answer as correct if it included a subset of the correct answer.

For the WikiDev3D, at least 50% of the subjects were able to answer 11 out of 13 questions. The subjects answered correctly 8.5 questions in average (st. dev. 3). As for the how easily they found the answer to the questions, the average score (in a range from 1 to 4) the was 3.38 (st. dev. 0.44) which corresponds to "fairly easily" qualitatively. For the 2D visualizations, at least 50% of the subjects were able to answer

11 out of 13 questions. The average number of correct answers was 9.75 per person (st. dev. 3.2). For the difficulty part, the average score for the WikiDev2.0 system was 3.43 (st. dev. 0.11). Examining each question individually we saw that for some of them the subjects gave a wrong answer in one system, but they corrected it in the other.

For the difficulty index, we grouped the questions in three categories: communication/collaboration, which is about relationships between users, contribution, about relationships between users and artifacts and history, which includes the notion of time. From the results we see that WikiDev2.0 is better on communication/collaboration and history (Table I), while WikiDev3D is better to answer questions about contribution (Table II). The reason for this seems to be the fact that WikiDev3D is more artifact centric and can capture more easily relationships between users and artifacts.

Moreover, we saw that in most questions where the subjects had trouble finding the answer in one system, it was easier for them in the other system. Therefore, it is becoming clear to us that the two systems can potentially complement each other. However, overall WikiDev2.0 seems to be a slightly easier platform. This can be attributed to the fact that WikiDev2.0 has a more intuitive and familiar interface (hyperlinks and web pages) compared to navigating in a virtual world. However, it was not clear what platform the users preferred since 2 users picked WikiDev2.0 as the easiest platform, while the other 2 chose WikiDev3D.

TABLE I
DIFFICULTY INDEX FOR THE 2D VISUALIZATIONS.

	S1	S2	S3	S4	**AVG**
communication/collaboration	3.75	3.5	3.67	3.25	**3.56**
contribution	3.6	3.6	3.4	2.8	**3.35**
history	4	2.75	3.25	3.5	**3.37**
AVG	**3.78**	**3.28**	**3.43**	**3.18**	**3.43**

TABLE II
DIFFICULTY INDEX FOR THE 3D VISUALIZATIONS.

	S1	S2	S3	S4	**AVG**
communication/collaboration	3.5	2.5	4	3.75	**3.43**
contribution	3.8	3.67	4	3.6	**3.8**
history	3.25	2	3.25	2.75	**2.91**
AVG	**3.51**	**2.72**	**3.75**	**3.37**	**3.38**

VII. SUMMARY, CONCLUSIONS AND FUTURE WORK

In this paper, we presented our recent work on evaluating two alternative visualizations, in 2D and 3D, developed in WikiDev2.0, our lightweight, MediaWiki-based collaborative software platform. Both visualizations are web-based; the 2D visualization is based on state-of-the-art interactive line-and-pie-charts frameworks, while the 3D visualization relies on the Open Wonderland virtual world. The former is accessible to individual users, while the latter enables the collaborative exploration of the data by multiple users logged in the same virtual world. Both visualizations can present information about software and communication artifacts, as well as their changes; the former depicts changes as a sequence of events (listed in a time-line or in subsequent rows of a table), where the latter uses animation to show the history of the artifacts changes. Our first pilot study was not conclusive with respect to which of the two visualizations is "better" at communicating information about the software project and which is "easier" to use. It appears that each has its merits and both appear as practically equally easy to use to our (admittedly) small set of subjects. It will be the objective of the next phase of our study to attempt to precisely characterize their differences and relative advantages and shortcomings. More specifically, we plan to examine whether collaborative project review in Open Wonderland is beneficial, as compared to two subjects synchronously examining the project data on a single machine.

ACKNOWLEDGMENT

The authors would like to acknowledge the generous support of iCORE, NSERC, and IBM.

REFERENCES

[1] S. A. Drummond and M. Devlin, "Software Engineering Students' Cross-Site Collaboration: An Experience Report," *Proceedings of The 7th Annual Conference of the ICS HE Academy Conference, Trinity College Dublin*, August 2006.

[2] P. Layzell, O. P. Brereton, and A. French, "Supporting Collaboration in Distributed Software Engineering Teams," *Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific*, pp. 38–45, 2000.

[3] K. Bauer, M. Fokaefs, B. Tansey, and E. Stroulia, "WikiDev 2.0: Discovering Clusters of Related Team Artifacts," in *CASCON 2009*, Toronto, Canada, November 2-6 2009.

[4] M. Fokaefs, B. Tansey, V. Ganev, K. Bauer, and E. Stroulia, "WikiDev 2.0: Facilitating Software Development Teams," in *CSMR 2010*, Madrid, Spain, 2010.

[5] J. I. Maletic, A. Marcus, and M. L. Collard, "A Task Oriented View of Software Visualization," in *Proceedings of the 1st International Workshop on Visualizing Software for Understanding and Analysis (VISSOFT '02)*, Paris, France, June 26 2002, pp. 32–40.

[6] C. Knight and M. Munro, "Comprehension with[in] Virtual Environment Visualizations," in *Proceedings of the 7th International Workshop on Program Comprehension*, Pittsburgh, PA, 1999, pp. 4–11.

[7] A. Marcus, L. Feng, and J. I. Maletic, "Comprehension of Software Analysis Data Using 3D Visualization," in *Proceedings of the 11th IEEE International Workshop on Program Comprehension (IWPC '03*, 2002, pp. 105–114.

[8] O. Greevy, M. Lanza, and C. Wysseier, "Visualizing Live Software Systems in 3D," in *Proceedings of the 2006 ACM Symposium on Software Visualization*, Brighton, United Kingdom, September 04-05 2006, pp. 47–56.

[9] R. Wettel and M. Lanza, "Visual Exploration of Large-Scale System Evolution," in *Proceedings of WCRE 2008*, 2008, pp. 219–228.

[10] Z. Xing and E. Stroulia, "Api-evolution support with diff-catchup," *IEEE Trans. Softw. Eng.*, vol. 33, no. 12, pp. 818–836, 2007.

[11] A. A. Martinez, J. J. Cosin, and C. Garcia, "A Landscape Metaphor for Visualization of Software Projects," in *Proceedings of the 4th ACM Symposium on Software Visualization*, Ammersee, Germany, September 16-17 2008, pp. 197–198.

[12] H. Koike and H. Chu, "How Does 3-D Visualization Work in Software Engineering?" in *Proceedings of the 20th International Conference on Software Engineering*, Kyoto, Japan, April 19-25 1998, pp. 516–519.

[13] Z. Xing and E. Stroulia, "The JDEvAn Tool Suite in Support of Object-oriented Evolutionary Development," in *ICSE Companion 2008*, 2008, pp. 951–952.

[14] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*. CRC Press, 2000.

[15] J. W. Sammon, "A nonlinear mapping for data structure analysis," *IEEE Transactions on Computers*, vol. 18, pp. 401–409, 1969.