SecondWATCH: a workspace awareness tool based on a 3-D virtual world

En Ye, Lev A Neiman, Matthew Sargent, Hiep Q Dinh, Chang Liu
School of EECS, Ohio University, Athens, Ohio 45701

ICSE 2009 VANCOUVER

Problem statement

- Software engineers spend about **70 percent** of their time on **cooperative activities** [1].
- **Awareness of others' activities** is essential in **facilitating coordination** in a team.
- **Awareness** - "an understanding of the activities of others, which provides a context for one's own activities" [2].
- Results of several field studies in large software companies such as Microsoft [3].
  - **coworker and artifact awareness are most common information needs for developers**
  - **Inadequate tool support to help acquire them**

Proposed Solution

- Build a workspace awareness tool (named **SecondWATCH**) based on **Second Life (SL)**, a 3-D online **Virtual World**.
- SecondWATCH informs developers of workspace awareness information by
  - Monitoring team members' activities on their **local workspaces, version control repository, and bug tracking system**.
  - Extracting and analyzing awareness info
  - Visualizing it **real-time in SL** as a **common view** shared by the whole team
- SecondWATCH provides three type of info:
  - **Coworker presence information:** developers' status, and which task they are working on.
  - **Real-time artifact information:** For each artifact, which developers are changing it locally.
  - **History artifact information:** For each artifact, how many revisions are contained, who has checked in most recently or most often, and how recently the latest change has been committed.

3-D city metaphor

- Based on city metaphor in the CodeCity project[4].
- Uses city buildings to represent files,
- Uses city district to represent folders,
- The layout of the city represents the overall file structure of the software project.
- The buildings shown as differently colored stacked cylinders stand on top of city districts shown as flat blue rectangles (with color saturation representing nested folders).

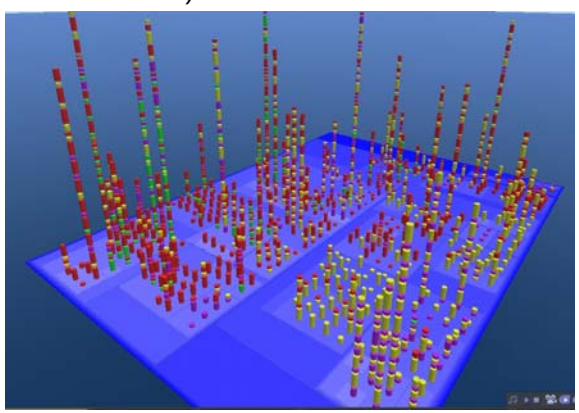

Figure 1. Visualization of the project Azureus

Table 1. Visualization mapping

Properties of a building	Properties of a software artifact
	History artifact information (static)
Height (Number of floors)	Number of revisions
Location of the floor	When the revision was checked-in (most recent revision always on top)
Color	The developer who commits this revision
Color transparency	How recent each revision was checked-in
	Real-time artifact information (dynamic)
Smoke emitting from the top of the building	Active artifact (Those artifacts that are being changed locally by at least one developer)
An icon rotating around the building	Developer who is making the local change
The building is shaking	More than one developers are changing the same artifact
The building with lights on (glowing)	Highlighted artifact

For more information:
http://vital.cs.ohiou.edu/vitalwiki/index.php/SecondWATCH

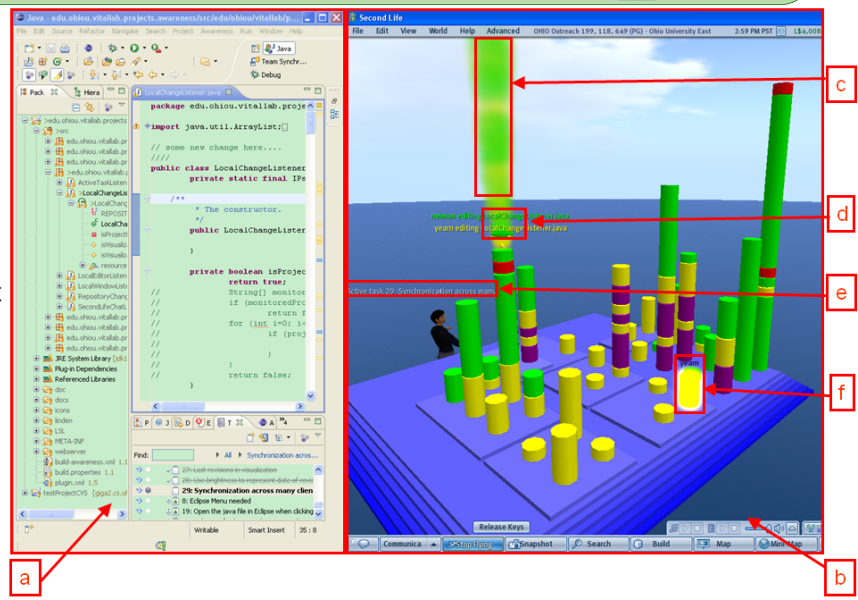

Figure 2. a) Eclipse b) Second Life c) Smoke representing active artifact d) Token representing the developer making local changes e) Active task f) Highlighted artifact

Implementation & Application

- Implemented as an **Eclipse plug-in** by integrating Eclipse with a modified SL client viewer.
- The info can also visualized as **a web page using Unity 3-D**
- Included **a filter** to filter out artifacts by **name, author, or check-in time** locally without affecting other developers' view
- Utilizes SL's **3-D object building feature** to create 3-D objects representing software artifacts, and its various **communication functionalities** to provide interaction between team members.
- Visualizes the history awareness information of **four open-source projects: FreeMind, jEdit, GUJ, and Azureus**

References

[1] Vessey and A. P. Sravanapudi. Case tools as collaborative support technologies. Communications of the ACM, 38(1):83–95, 1995.
[2] P. Dourish and V. Bellotti. Awareness and coordination in shared workspaces. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 107–114, New York, NY, USA, 1992. ACM Press.
[3] A. J. Ko, R. DeLine, and G. Venolia. Information needs in collocated software development teams. In *ICSE '07: Proceedings of the 29th International Conference on Software Engineering, pages 344–353, Washington, DC, USA, 2007.* IEEE Computer Society.
[4] R. Wettel and M. Lanza. Program comprehension through software habitability. In *ICPC '07: Proceedings of the 15th IEEE International Conference on Program Comprehension*, pages 231–240, Washington, DC, USA, 2007. IEEE Computer Society.