

Introduction

Virtual world environments afford new possibilities for the application in the field of software engineering. There is a great demand to support awareness across distributed development teams.

From a theoretical design perspective, based in the field of Human-Computer Interaction (**HCI**), virtual design environments present an interesting opportunity to re-evaluate the relationship between designer and end-user.

Motivation

Distributed software development on a global scale has become common practice. There is a growing demand for visualization tools that enable developers and project managers to collaborate with distant team members in efficient and productive ways.

Awareness is necessary on various levels of the software design process. Initial, **high level processes** like brainstorming discussions, UML modeling and use case diagrams are common group practices. Ideas and results need to be collected and stored in a central place that is accessible to all distributed team members.

Once the project approaches the actual implementation stage, the distributed team members focus on **low level design processes** like design patterns, data models and API usage.

Most software design visualizations tools available today often only provide awareness on a relatively deep and isolated software design level. What is missing is an approach that provides a more **coherent awareness state across high and low level design process phases**.

Tracking back responsibilities and comparing former design decisions in distributed teams is difficult because a **persistent design space** is necessary that can both represent actors and shared objects in an intuitive way.

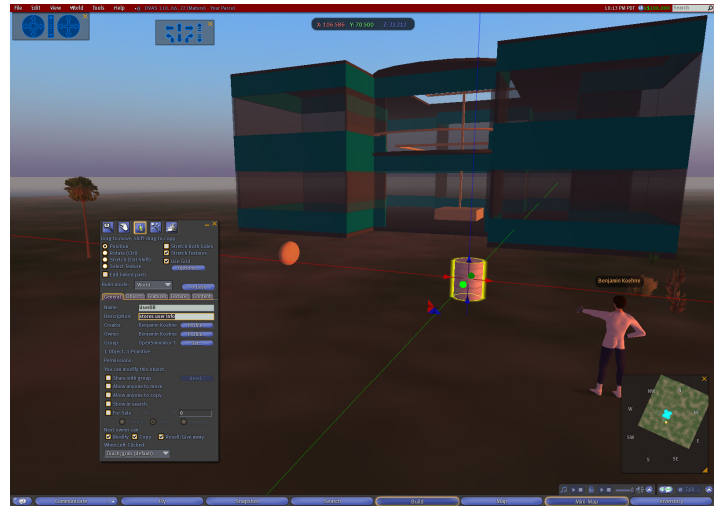


Figure 1: Virtual Design Environment

Approach

A virtual world environment serves as a persistent virtual design space that allows data flow between the virtual collaborative awareness environment and an IDE.

- Directly linked to IDE (switching between IDE and virtual environment, automated data flow)
- Regions and islands represent design stages (brainstorming, modeling, implementation)
- Avatar's location and appearance signal responsibility and current activity
- Components can be exported as tangible objects and shared with team members
- Versioning allows going back to early drafts and design sketches
- Design and implementation activities become expressive in the virtual environment

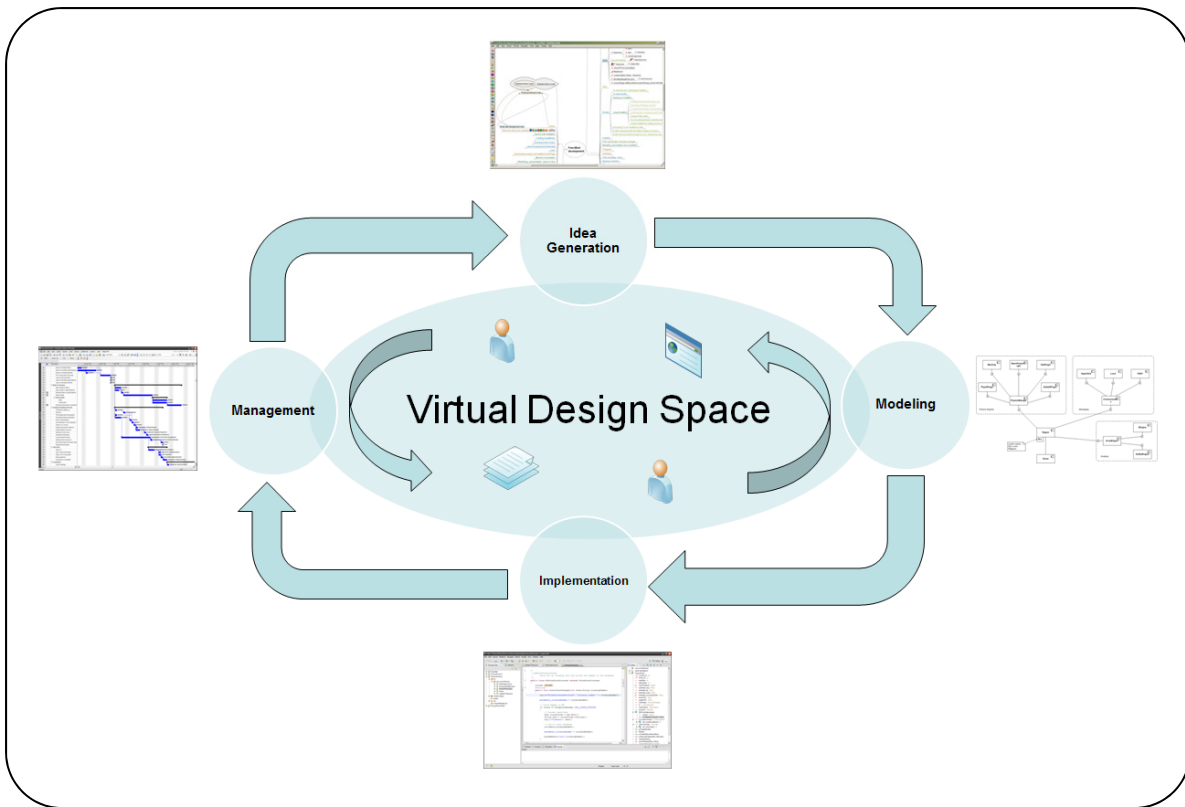


Figure 2: Embodied Software Design Process

Main Benefits

- **'Feel' for Components:** Dependencies, contributors and history
- **Automated Data Flow:**
No considerable overhead for developers
- **Observer Mode:**
Explore software project from different levels of abstraction
- **Flexible Awareness:**
Helpful for designers, coders and managing parties
- **Rethinking the Designer- / User- Image**
Allows reflection on the roles of end-user and designer

Conclusions

Embodied Software Design Spaces contributes with a new approach for distributed software development. A persistent, yet virtual, design space provides awareness across development stages and embodied components.

The system is suitable for globally distributed development teams that require a coherent awareness environment linking to standard development tools such as Eclipse.

Virtual world environments present new ways to think about design practice in general. End-users and designers become part of a continuous information sharing process.

Contact Information

Advisor: Professor David F. Redmiles
 Project: Professor André van der Hoek
 Benjamin Koehne, PhD student

Donald Bren School of Information and
 Computer Sciences
 Department of Informatics
 University of California
 Irvine, California 92697-3425

{redmiles, andre, bkoehne}@ics.uci.edu