Managing Software Projects in Spatial Hypertext: Experiences in Dogfooding

Frank Shipman
Center for the Study of Digital Libraries and Department of Computer Science
Texas A&M University
College Station, TX 77843-3112, USA
shipman@cs.tamu.edu

Abstract

Managing long-term, research-oriented software projects requires more flexibility and open-endedness than most production-oriented software processes provide. We have been exploring the use of spatial hypertext to manage such projects. Spatial hypertext allows users to place information objects in visual spaces and use visual cues and spatial relations to represent inter-object relations. Over time, users develop a visual language to express characteristics of their task. The Visual Knowledge Builder (VKB), our particular spatial hypertext system, uses heuristics to recognize structure in user-generated layouts and includes navigable history for returning to earlier states of the spatial hypertext. This paper reflects on our experiences in dogfooding – using our own research prototype – for two projects for more than two and a half years and what these experiences might mean for using spatial hypertext in other software development contexts.

Keywords

spatial hypertext, software development

1. Introduction

Software development takes place in a wide variety of contexts, yielding an equally wide variety of software engineering processes. Some contexts, such as life-critical applications, require stability and correctness and are willing to expend resources (including delaying delivery time) in order to achieve these goals. Other contexts, including much software developed for home use, are time-to-market driven, looking to gain market share with the potential of providing bug fixes later and enhancements through a series of versions of the software. Research software is even more extreme – the software is being developed to explore what is possible within a design space. In the research context, reliability need only support the project’s mode of evaluation, e.g. proof-of-concept demonstrations, laboratory experiments, and limited real-task usage being common.

Different project management software is appropriate within these different contexts. We are exploring the use of spatial hypertext, with its emphasis on free-form expression, in the project management of research software. Spatial hypertext users place information objects in visual spaces and use visual cues and spatial relations to represent inter-object relations [4]. For example, they may categorize objects by creating lists or using color. Over time, users develop a visual language to express characteristics of their task. Our system, the Visual Knowledge Builder (VKB), includes a spatial parser for recognizing structures in the layout and supporting users in the manipulation of these structures [7]. Additionally, VKB records the evolution of the workspace as a navigable history with multiple access mechanisms.

The next section provides a brief overview of VKB, its spatial parser, and its history mechanism. Following this is a description of our use of VKB for managing software-oriented research projects. The paper concludes with a discussion of how our experiences might inform the use of spatial hypertext in other software development contexts and the development of project management software.

2. The Visual Knowledge Builder (VKB)

The main Visual Knowledge Builder interface, shown in Figure 1, is a two-dimensional workspace with controls at the top and message bars below. Users collect and author
information in the form of rectangular objects containing text, attributes and values, links to files or URLs, and images. User’s express interpretations, e.g. categorizations and relations, through the placement of objects and the use of visual attributes, such as border and background color, border width, and font type, size, and color. The workspace also includes “collections”, two-dimensional spaces embedded in the top-level workspace or another collection. Users may navigate into a collection to see more of its contents. Figure 2 shows the results of navigating into the “To Do List” collection in Figure 1. For more information on general VKB functionality see [7,8,10].

Figures 1 and 2 show a workspace used for managing a research project with five or more participants at any one time. This space has been in use for over two years in weekly project meetings. Writing tasks are placed in the “Paper topics -- areas for work” collection and system and design tasks are found in the “To Do List” collection. Over the period of use, dozens of tasks have been identified, given a priority, and placed in the “Done” collection on the right side of Figure 2.

The second toolbar in Figures 1 and 2 provides access to the history of the workspace. VKB records all the editing events and allows users to play this record forwards or backwards, navigate to specific types of events, or locate the state of the workspace on a particular date [6]. This history mechanism is similar to Reeves’ embedded history [5] and Hayashi’s temporally-threaded workspaces [1].

2.1 Structure Recognition

To aid the manipulation and later formalization of visually-represented information, VKB attempts to recognize spatial structures as they are created in the workspace via a spatial parser. The spatial parser was developed to recognize structures found to be common in a variety of virtual and physical layouts [9], such as the lists, stacks, and composites in figures 1 and 2. Users can access different scopes of structure through hierarchic-click selection and the recognized structures are used to generate suggestions for placement, formal attributes, and relations of information objects [10].

Figure 1. Project workspace with three collections containing software development tasks, paper-writing tasks, and brainstorming for a particular software extension.
Figure 3 shows a limitation in the spatial parser. Much like the space seen in figures 1 and 2, this is a project workspace used in weekly meetings. Unlike the arrangement in the prior space, this VKB space uses horizontal position to indicate a continuum of priority. The spatial parser recognizes (some) horizontal lists in this structure but does not include the notion of an ordered list, much less the use of space to represent continuous values of an attribute such as priority. Techniques from Hsieh’s VITE [2,3], which supports continuous and discrete mappings between structured data and a visual representation, could aid in this example.

### 2.2 Navigable History

One feature of VKB that has been particularly useful within the context of project management is the navigable history. The initial to-do lists had relatively few tasks but have grown as dozens of tasks being generated (and fewer being completed) during each year. During this time, the

---

**Figure 2.** Navigating into “To Do List” collection in Figure 1 shows lists indicating the priority of tasks, border colors indicating person responsible for task, and border width indication progress on task.
visual representation – the semantics of different colors and border widths – has evolved to cope with more tasks and to represent characteristics of tasks considered important later in the project’s development. As such, the visual attributes of older tasks (particularly those in the “Done” or “Completed” collections) cannot be interpreted based on the current meaning of those attributes. Navigable history allows returning to earlier states of the workspace to determine the meaning of the visual representation. Also, by going back to prior times in the workspace, the creation and modification of tasks can be viewed. Figure 4 shows the workspace from Figure 3 approximately 1 year earlier.

3. Experiences from Dogfooding

Dogfooding is the use of one’s own systems for debugging and iterative design. It is not a substitute for getting users involved in design and formative or summative evaluation. We have been using versions of VKB for note taking, preparing papers and presentations, project management, and conference organization. VKB has also been in use by members of the hypertext research community (outside of Texas A&M) for over two years.

Our experiences using VKB to help manage the research projects shown in the above figures began in November, 1999. The VKB spaces are projected and edited during weekly meetings. Because the project team members (faculty and students) are in the room during most editing of the space, the implicit nature of the representation remains comprehensible. The face-to-face setting promotes the use of conversation to repair breakdowns when visual changes are not understood by all participants.

The visual languages have also become the focus of humor. For example, border color in Figures 1 and 2 indicates who has primary responsibility for each task. When new tasks are created there can be a variety of suggestions as to what color it should be assigned. As border thickness indicates progress, there is competition surrounding the changing of task borders and movement of tasks to “Done”. Finally, for some of the most dreaded tasks – like writing journal papers – the font gets bigger each week until some progress is achieved. Clearly, group personality plays a large role in such activity. While we
have not performed any comparison, it seems less likely that such good-natured banter and competition would occur using a traditional project management tool to track progress. The implicit nature of the visual representation makes these judgments less threatening.

Besides promoting a common understanding of tasks and progress, the workspaces act as a community memory of project activity. When an annual report is due, going back through the history helps identify activities to report. Seeing the edits replayed triggers memories of not just what was recorded but more general meeting discussions that occurred. Another use of the history has been to determine when particular ideas or tasks were introduced into the project. Thus the workspace and its history act as a record of the intellectual development of the projects.

4. Discussion

To determine how our experiences with the use of spatial hypertext for project management can inform its potential for use in other software development contexts, we must first understand the similarities and differences in these contexts. Large research software projects are characterized by their continual exploration of a design space. This exploration is driven by the interests of the members of the project, the availability of funding, the research methods, and the venues for reporting results.

A successful project is one that generates publishable results, which often includes developing software prototypes that can be studied in use. A really successful project uncovers new issues that cause the project to continue indefinitely.

While this may seem very different from commercial software development, there are a number of similarities. The continual integration of design and development can be found in many iterative development methods. Also, most software developed that is successful continues to evolve for years after the initial release. No one believes the current version of MS Word will be the last. Thus, potential for indefinite future development is also of value for commercial software. Many of the differences between commercial and research software seem to be in the required level of stability, features, support, and scale.

The primary advantage to using spatial hypertext is its permissive nature. The visual representation allows expression of characteristics of the task as desired. Schemas do not have to be changed when new it is determined that new attributes of tasks need to be
considered. Also, the informal nature of the representation removes the need to have separate tools for recording design discussions and managing the software development.

The informal nature means that this would not work in environments where there is the need to automatically generate reports on the status of projects. Also, while there can be an indefinite number of objects in a spatial hypertext, limitations of displays pose practical problems for working with more than a few hundred tasks in one space. Thus, spatial hypertext seems well suited to relatively small software development activities that require, or would benefit from, flexibility.

Looking to the future, spatial hypertext could better support our own development in a couple ways. The most obvious direction is connecting the visual workspace with structured or semi-structured data used by other tools. Integrating VITE’s ability for visual changes to objects in the workspace to be reflected as semantic changes to objects in an underlying database would allow integration with other software development and project management tools.

5. Acknowledgements

This work was supported in part by the National Science Foundation under Grant Number IIS-9734167.

6. References