Requirements Elicitation using Visual and Textual Information

J. Michael Moore, Frank M. Shipman III
Department of Computer Science
Texas A&M University
TAMU 3112
College Station, TX 77843-3112 USA
1 979 845 5534
Email: {jmichael, shipman}@cs.tamu.edu

1. End Users as a Source of Requirements Data

Many techniques are available for elicitation and refinement of requirements [5]. However, these require face-to-face interactions that are expensive in time and often require co-location. For these reasons, crucial interactions between end-users and requirements analysts may not even happen. In these situations, other techniques that do not require face-to-face interactions, such as questionnaires, are used.

Combining communication with reference to an artifact can facilitate communication by allowing "design by doing" [1]. Moreover, Glenberg and McDaniel have noted that integrating spatial and linguistic information is a requirement for effective communication [2]. Others have also used a combination of visual and textual information. HOSS uses visual design artifacts as the focus for communication [7]. VKB uses visual objects for information organization and interpretation [8].

2. Approach

Our approach is to provide a software tool that capitalizes on the "language of the GUI," combining visual and textual information synergistically for communication facilitation.

The use of graphical user interfaces (GUIs) has become ubiquitous through the use of operating system interfaces such as Microsoft Windows and the Macintosh OS. Through this interaction paradigm the usage of widgets has become standardized to the point where the placement and usage of widgets are essentially a visual language, the language of the GUI.

By using this language, interface construction becomes a way for end users to communicate their desires and goals for software. End users can convey things that might be too difficult using solely text. Information conveyed only with graphical constructions may be insufficient for adequate communication. These graphical artifacts can be clarified by attaching textual argumentation, allowing the strengths of both textual and visual information to augment communication.

The language of the GUI allows communication of requirements through end user designs. This is viewed as design for elicitation rather than a final design.

End users with domain expertise often know the information that a system needs to deal with but do not know how to convey this knowledge [6]. Consequently, they give more high-level functional descriptions and omit what could be crucial details. Grounding end users in the environment of the GUI helps them to think more specifically about tasks and the specific details related to the transfer of information into and out of a computer.

This approach provides an alternative to questionnaires when face-to-face communication is infeasible. It is intended as a tool for initial requirements gathering.

This approach could be applied to sketches of interfaces or to interfaces built in GUI building environments. However, the GRC environment (described below) promises the ability to automate many time consuming aspects of the analysis.

3. Graphical Requirements Collector (GRC)

The Graphical Requirements Collector (GRC) is a software tool for gathering requirements directly from end users [4]. End users are instructed to create their vision of an interface for a specific software application.

As GRC users create interface constructs they are conveying information in two ways. First the construction of windows and the organization of widgets provide information through the language of the GUI (Figure 1). Argumentation can be associated with widgets regarding

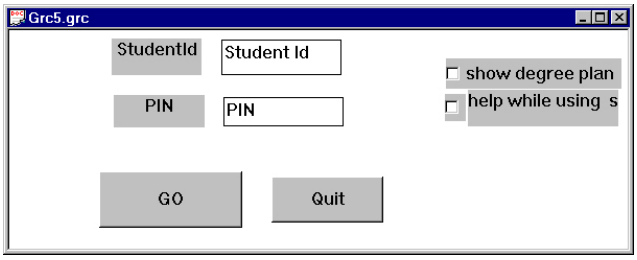

Figure 1: End user interface construction

what they represent or do along with rationale for why the widgets were created. In this way users can take advantage of both textual and visual information, providing it with the mode that best expresses their intentions [8].

GRC is being developed as a two part system. The first gathers interface constructions and argumentation from end users. The second is a tool for software engineers to use for analysis and management of end user generated constructions. It will assist with the extraction of rudimentary models and requirements data that can be extended into a complete requirements set using other established techniques.

4. Study of GRC Constructions

Six university students participated in a study to investigate requirements gathering using GRC. The students constructed an interface for an online course registration system and attached textual argumentation.

4.1. Textual Data

The textual data collected with GRC was compared with data collected using a more traditional questionnaire [4]. Surprisingly, it was found that GRC garnered more textual expression among subjects than the questionnaire did. Other results included that questionnaires produced more high level features where GRC produced more fine grained behavioral information. Our previous work focused on the textual argumentation with little examination of the graphical constructs.

4.2. Graphical Data

Analysis of the graphical data was insightful. User constructions generated through GRC provided information that supports existing requirements engineering and refinement techniques. Types of information identified include: objects, classes, activity diagrams, use cases, constraints and rationale. In no case was the information complete, but provided a starting basis for additional exploration and refinement. The poster illustrates information extraction from the data.

The graphical nature of the data lends itself to object oriented analysis. Moreover, graphical constructions often translate easily into graphical diagrams such as class diagrams and activity diagrams. Translation is facilitated by the fact that graphical constructions often have a one to one correspondence to pieces in these diagrams.

The analysis provides insight into how these cases can be generalized for automated extraction, the next major phase for developing this tool into something beyond a rudimentary GUI design tool.

4.3. Combined

Limiting users' expressions to only one medium (as with questionnaires) may result in failure [3]. The use of a graphical component mixed with textual argumentation provides people options for expressing knowledge. Knowledge that was inexpressible through text is conveyed graphically and vice versa. When neither is expressive enough alone, they can be combined to convey together what was difficult or impossible separately.

References

[1] P. Ehn, Playing the Language-Games of Design and Use-on Skill and Participation, *ACM Conference on Supporting Group Work*, 1988, pp. 142-157.

[2] A.M. Glenberg and M.A. McDaniel, Mental models, pictures, and text: Integration of spatial and verbal information, *Memory and Cognition,* 20(5), Psychonomic Society, Inc., 1992, pp. 458-460.

[3] J.A. Goguen and C. Linde, Techniques for Requirements Elicitation, *1993 Proc. of IEEE International Symposium on Requirements Engineering*, 1993, pp. 152 –164.

[4] J.M. Moore and F.S. Shipman, A Comparison of Questionnaire-Based and GUI-Based Requirements Gathering, *Proc. ASE 2000, The Fifteenth IEEE International Conf. on Automated Software Engineering*, 2000, pp. 35-43.

[5] B Nuseibeh and S. Easterbrook, Requirements Engineering: A Roadmap, *ICSE-2000 The Future of Software Engineering, A. Finkelstein (ed),* 4-11th June 2000.

[6] Polanyi, M., *The Tacit Dimension*. Garden City, NY: Doubleday, 1966.

[7] B. Reeves and F. Shipman, Supporting Communication between Designers with Artifact-Centered Evolving Information Spaces, *Proc. CSCW '92, ACM 1992 Conf. on Computer-Supported Cooperative Work,* 1992, pp. 394-401.

[8] F.M. Shipman, H. Hsieh, R. Airhart, P. Maloor, J.M. Moore, D. Shah, Emergent Structure in Analytic Workspaces: Design and Use of the Visual Knowledge Builder. To appear in Proc. of Interact 2001.