A Comparative Study of the EUREQA Tool for End-User Development

Paul G. Austrem, University of Bergen and Webstep AS, Norway

ABSTRACT

End-user development is a growing domain; however, few dedicated end-user development tools exist allowing end-users to incorporate their domain knowledge into software solutions. EUREQA is a design pattern driven UML class diagram modeling tool allowing end-user developers to create UML class diagram models reflecting their domain knowledge. The UML class diagram can be used as a basis for model-driven development. EUREQA was evaluated with 20 participants recording data through direct observation, screen recordings, the think-aloud protocol and semi-structured interviews. Half of the participants used EUREQA and the other half used the Microsoft Visio modeling tool. The authors found that EUREQA participants scored higher than Visio participants for this task and that skill had less effect on the EUREQA participants’ solutions. Specifically inexperienced EUREQA users struggled less when working with UML class diagram elements compared to Visio participants.

Keywords: Design Patterns, End-User Development, Information Systems, Object-Oriented Modeling, Tool Support

INTRODUCTION

End-user development is a growing domain, Scaffidi et al. (2005) state that there in the near future will be only 2.5 million professional software engineers compared to 55 million end-user developers in the United States. Survey results from McLeod, MacDonnell, and Doolin (2007) reported that user participation occurred in parts of IS development projects 92% of the time and 62% in the entire project. Thus user participation and end-user development is a major aspect of real-world IS development. However, there are few dedicated end-user development tools available to support this.

The EUREQA (End-User Requirements-based dEvelopment with Quality Attributes) tool offers an environment allowing end-user developers with basic software modeling skills to work at the model level and use design patterns as building blocks to construct a UML class diagram model. The purpose of EUREQA is to enable a specific segment of end-user developers to create UML class diagrams which reflect their domain expertise without burdening them with learning the UML modeling language’s syntax and notation. This is accomplished through piecemeal assembly of design patterns to cre-
ate UML class diagram models. The intended purpose of the UML models are as skeleton solutions representing a high-level UML model description of a systems behavior through design patterns, domain knowledge through named UML class entities and establishing a shared vocabulary for end-users and developers for further development into a running solution.

In this paper we present a comparative user evaluation of EUREQA and the Visio modeling tool. The following section will discuss the background and theoretical foundation of EUREQA. This will be followed by a description of the research questions, a method and data collection section, the analysis and a discussion of the findings. Finally conclusions and further work is presented.

BACKGROUND MATERIAL AND THEORY

End-User Development

Lieberman et al. (2006) define end-user development as “a set of methods, techniques, and tools that allow users of software systems...to create, modify, or extend a software artefact” (p. 2). The definition shows the breadth of what can be considered end-user development. Within their definition EUREQA belongs to the tool dimension.

Lieberman et al. (2006) also provide a further refinement of tool-based end-user development with four distinct approaches, *programming by example, incremental programming, model-based development, extended annotation or parameterization*. Model-based development is the richest of the four approaches at the cost of being the most complex. The richness and freedom allowed in a tool can be considered directly proportional to its complexity. In end-user development it is a design goal to create an environment with a flat learning curve whilst still affording the end-user developer the freedom to achieve their goals.

End-User Development Tool Heuristics

The users of end-user development tools have a very different profile compared to professional software developers. They do not have deep knowledge of software engineering and its principles, nor do they have a desire to dedicate large amounts of their time to learning to use development tools. Additionally they may not have the understanding of professional developers who can easily acquaint themselves with new software and tools. This has lead to normative research into programming languages and systems for end-user developers. Green and Petre’s (1996) work on developing tool design heuristics for novice programmers has served as guiding in the development of EUREQA. They propose heuristics that should be considered when developing a programming system for novice users. The following section presents the theoretical foundation that formed the development of EUREQA with a basis in Green and Petre (1996).

Consistency with Metaphor

These deal with providing the end-user with an environment that they can relate to, the semantics of the language should represent the underlying model in a way that is understandable for the end-user developer. It should represent some real-world artifact that the end-user can use to understand what a system component does. “Novices expect the computer to understand a variable based on its name” (Pane & Myers, 1996, p. 17). EUREQA deals with this issue by leveraging design patterns and states the purpose of each class in a design pattern allowing the user to change the class name to reflect their domain.

Consistency with External Knowledge

Costabile et al. (2008) state that end-user developers often are forced to follow the mental
Tourists’ Mobile Information Seeking Behavior: An Investigation on China’s Youth

Teaching Software Engineering in a Computer Science Program Using the Affinity Research Group Philosophy