Chapter 98
Evaluating the Usability of Domain-Specific Languages

Ankica Barišić
Universidade Nova de Lisboa, Portugal

Miguel Goulão
Universidade Nova de Lisboa, Portugal

Vasco Amaral
Universidade Nova de Lisboa, Portugal

Bruno Barroca
Universidade Nova de Lisboa, Portugal

ABSTRACT

Domain-Specific Languages (DSLs) can be regarded as User Interfaces (UIs) because they bridge the gap between the domain experts and the computation platforms. Usability of DSLs by domain experts is a key factor for their successful adoption. The few reports supporting improvement claims are persuasive, but mostly anecdotal. Systematic literature reviews show that evidences on the effects of the introduction of DSLs are actually very scarce. In particular, the evaluation of usability is often skipped, relaxed, or at least omitted from papers reporting the development of DSLs. The few exceptions mostly take place at the end of the development process, when fixing problems is already too expensive. A systematic approach, based on techniques for the experimental evaluation of UIs, should be used to assess suitability of new DSLs. This chapter presents a general experimental evaluation model, tailored for DSLs' experimental evaluation, and instantiates it in several DSL's evaluation examples.

INTRODUCTION

Software Languages Engineering (SLE) is becoming a mature and systematic activity, building upon the collective experience of a growing community, and the increasing availability of supporting tools (Kleppe, 2009). A typical SLE process starts with the domain engineering phase, in order to elicit the domain concepts. The next phase consists in the actual design of the language, by capturing the referred concepts and their relationships. This is followed by its implementation, evaluation, deployment, evolution, and finally its retirement. Although this process is becoming streamlined, it still presents a serious gap in what should be a crucial phase - *language evaluation,* which includes acceptance testing. A good DSL is hard to build because it requires domain knowledge and language development expertise, and few people have both (Mernik, Heering & Sloane, 2005). We

DOI: 10.4018/978-1-4666-4301-7.ch098

should evaluate claims such as *"our new language brings efficiency to the process"*, or that *"our new language is usable and effective"*, with an unbiased and objective process.

DSLs are meant to close the gap between the Domain Experts and the computation-platforms. As such, DSLs can be used as a structured/comprehensive means to achieve Human/Computer (H/C) Interaction. Most of the requirements concerning the evaluation of User Interfaces (UI) are actually associated with a qualitative software characteristic called *Usability*, which is defined by the quality standards in terms of achieving the Quality in Use (ISO, 2001a).

Usability evaluation involves a phase of acceptance testing with actual users, which is typically a very costly process. A poorly conceived evaluation process can ultimately undermine the conclusions about the quality of the UI under analysis. This generic UI problem also applies to the realm of DSL's construction.

In our opinion, usability can be fostered from the beginning of the DSL development cycle by adopting user centered methods. The objective is to ensure that the developed DSLs can be used by real people (the domain experts) to perform their tasks in the real world. This requires not only intuitive UIs, but also the appropriate functionality and support for the activities and workflows that are to be specified with the DSLs.

In this chapter, we discuss how user-centered design can be adapted to the context of DSL's development. In general, working with languages involves not only physical and perceptual activities, but also cognitive activities such as learning, understanding and remembering. Experimenters in human factors have developed a list of tasks to capture those particular aspects. The process is complex and must be tailored case-by-case (Reisner, 1988). We will further discuss these issues, and show how they fit into the DSL's development process. Following that, we will define a general model for the DSL's experimental evaluation. This model will help us planning and designing the DSL's evaluation processes, as well as conducting post-mortem analysis of other DSL's evaluation efforts in a systematic way, thus fostering the aggregation of several DSL's evaluation results. As discussed in (Basili, 2007), a single study outside the context of a larger set of studies has limited value, but combined, they can be a valuable increment to the existing body of knowledge.

The usage of our model is illustrated through the systematic analysis of several evaluations of DSLs found in the literature. Our comparative analysis will help identifying the commonalities, differences, strengths and weaknesses of the compared studies. The usage of our model in future replications of this comparative study to other DSL evaluations has the potential for fostering meta-analysis, leading to sound increments of the body of knowledge in DSLs and their evaluation.

BACKGROUND

In general, the software industry does not report investment on the usability evaluation of DSLs, as shown in a recent systematic literature review (Gabriel, Goulão & Amaral, 2010). This conveys a perception that there is an insufficient understanding of the SLE process which, in our opinion, must include the evaluation of the produced DSLs. Many language engineers may perceive the investment in usability evaluation as an unnecessary cost and prefer to risk providing a solution which has not been properly validated, namely with respect to its usability, by the end users. This apparent state of practice contrasts with the return of investment on usability reported for other software products (Nielsen & Gilutz, 2003). In general, these benefits span from a reduction of development and maintenance costs, to increased revenues brought by an improved effectiveness by the end users (Marcus, 2004).