A Scenario Construction Process

Julio Cesar Sampaio do Prado Leite\textsuperscript{a}, Graciela D. S. Hadad\textsuperscript{b}, Jorge Horacio Doorn\textsuperscript{c} and Gladys N. Kaplan\textsuperscript{d}

\textsuperscript{a}Pontificia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brazil; \textsuperscript{b}Universidad Nacional de La Plata, Buenos Aires, Argentina; \textsuperscript{c}Universidad Nacional del Centro de la Provincia de Buenos Aires, Buenos Aires, Argentina; \textsuperscript{d}Universidad de Belgrano, Buenos Aires, Argentina

Scenario is a description technique that has been attracting a lot of attention from practitioners and from researchers. Several disciplines have been using scenarios for some time now, but recently the information system community has dedicated special attention to the possibilities that this description technique provides to enhance understandability of task-related descriptions and communicability among stakeholders. This paper aims its attention at a particular scenario construction process, but while doing so it tackles important problems regarding scenario management, in particular scenario organisation. Our scenarios are described in a structured way, using a simple conceptual model together with a form-oriented language. We have been using this representation for some time now, and our results are based on several case studies conducted with real-world problems.

Keywords: Organisational context; Requirements elicitation; Scenario management; Scenario modelling; Stakeholders

1. Introduction

The necessity of ensuring good understanding between requirements engineers and clients/users has motivated the research and development of methods that allow better results in the collaboration among all participants in the requirements definition process. Requirements engineers should understand, model and analyse the application domain where the software will run and the clients/users must validate whether or not the engineers’ vision is correct. Our research uses the idea of scenario as a medium to achieve this goal.

The word scenario, defined as ‘the plot of a motion picture’ in the Merrian–Webster Dictionary, has been used for a long time in several areas of different disciplines (military strategy, decision making, marketing, economy). The information system area started using scenarios in the human–computer interface and recently in software engineering. Jarke, Bui and Carroll [50] provide an overview of scenario research in a broader view, both from an information system point of view as well as from a management (decision theory) view. They point out that the effectiveness of the use of scenarios in several disciplines is fundamentally due to their capability of stimulating thinking. Scenario provides a situated task vision together with an effective way of communication among the actors involved in the subject of study.

For us, scenarios describe situations that happen in the Universe of Discourse (UofD).\textsuperscript{1} Carroll [5], taking into account usage aspects, believes that scenarios allow us to know the problem, to unify criteria, to gain clients/users compromise, to organise the involved details and to train new participants. The use of scenarios, as a way of understanding the domain in which the software will perform, has been recommended by several authors [6–9] and those proposals became very important for extending the use of scenarios in real practice. However, a detailed analysis of the recommendations from the

\textsuperscript{1}The overall context in which software will be developed and operated. The UofD includes all the sources of information and all the people related to the software. It is the reality trimmed by the set of objectives established by those demanding a software solution.’ [1, 2]. We use the term Universe of Discourse with the same meaning given by Michael Jackson [3] to Application Domain. Loucopoulos [4] also uses both terms as synonyms.
literature shows some degree of dispersion and contradiction in the use of scenarios. Recent work of the CREWS project confirms this observation [10,11].

This lack of precision of when and how scenarios should be used has spread to the engineers who are using these techniques in the field. Thus, most developers see scenario creation more as a craft than as an engineering task. Recent studies concerning the use of scenarios in industrial projects [11] have clearly proved this fact and pointed out the necessity of more detailed definitions on scenario construction to increase their use in real situations. This is in complete agreement with Rolland et al. [12] and Sutcliffe [13], who think that there is little understanding about the usage and production of scenarios.

The variety of interpretations, syntax and construction mechanisms for scenarios goes as far as showing up basic contradictions. For instance, with respect to the scenario construction process there is no agreement whether scenarios should be built in a top-down or in a bottom-up fashion. We believe that part of the problem of such diversity in the scenario literature is due to the fact that the scenarios building process should be neither top-down nor bottom-up.

The present work proposes a strategy for the creation and use of scenarios as a grounding representation for the requirements engineering process. The strategy is based on the assumption that scenarios must rely on natural language as a means of communication among stakeholders, in particular among clients/users and requirements engineers. The strategy also relies on the idea of scenario evolution [14], starting from what Rolland et al. classified as organisational context [10], which aims at the ‘broad picture of how the work gets done’ [10, p. 30], and what Jarke et al. [50] called environmental scenarios. Our scenarios are neither specifications nor requirements, they are auxiliary descriptions for the process of requirements definition. They provide a knowledge source where requirements may be found and specifications may be based upon.

This paper details a scenario construction process. This process addresses two important problems related to scenario management [50]: organisation and quality. The organisation problem arises as we start to deal with a large number of scenarios. The quality problem refers to the reliability of the descriptions presented as scenarios. Regarding the first problem, we provide an innovative middle-out strategy that systematises the construction process using typed relationships and operational heuristics. Regarding quality, we provide policies and procedures for detecting defects and errors in scenarios. It is important to stress that our proposal is based on a cumulative experience on the use of scenarios. For the last four years, we have analysed more than 20 software projects that used scenarios in the requirements process. In total we have analysed more than 400 different scenarios.

We understand that describing the process, giving data on its use and comparing it with other work contributes to increase the knowledge about scenario management. We have organised the paper into six sections. Section 2 provides a survey of existing scenario construction processes. In section 3 we present our strategy, giving emphasis to its reliance on natural language. Detail on how scenarios are put together is given in section 4. Section 5 reports on our experience in producing scenarios. In conclusion we state that our results are important to scenario management by comparing our work with previous work, as well as pointing to future research.

2. Different Approaches for Scenario Construction

There are several styles in which scenarios are built. such as textual narrative, storyboards, video mock-ups and written prototypes [5,7,8,9,15]. This section stresses the distinction between top-down and bottom-up strategies by quoting some of the construction heuristics proposed by the literature. We believe that this review is important since our proposal, a middle-out strategy, is perceived by us to be a major contribution. This belief is based on our own experience of previously using top-down and bottom-up strategies. In the following paragraphs we present different quotations from the literature.

Even if it is not clearly stated, Booch [16] seems to adhere to a top-down approach since: ‘The most complex application can be characterized in terms of a few dozen primary scenarios. Primary scenarios model the central problem. A Primary scenario represents some fundamental system function. Secondary Scenarios represent some variation on the theme of a primary scenario. The entire desired behavior of a software system can be captured through a web of scenarios, much the same way as a storyboard does in making a movie.’

Booch’s primary scenarios are seen as relevant scenarios by Firesmith [17]; they are linked together by a scenario lifecycle diagram: ‘A scenario lifecycle diagram is used to document the valid interactions