

Functional and Non-functional Requirements Specification for Enterprise Applications

Renuka Sindhgatta and Srinivas Thonse

Infosys Technologies Limited,
Bangalore, India, Phone 91- 80-5117 3908
{renuka_sr, srinit}@infosys.com
<http://www.infosys.com>

Abstract. Comprehensive and accurate software requirements capture is essential for successful development of software systems. Enterprise applications have an additional challenge of eliciting requirements that need to be well understood by i) the business users of the system having extensive domain knowledge ii) application developers having extensive system implementation and development knowledge. Current tools vary from providing textual descriptions to formal semantic languages for specifying requirements. The business users are unable to actively participate in the analysis, as formal and textual specifications represent two extreme ends of requirements elicitation. Ambiguity or lack of understanding often poses a challenge on validation and verification of system requirements specification. The paper presents a Use case Specification Framework that brings structure to requirements specification while retaining its simplicity. The framework enables business users to understand and verify functional requirements and two critical non functional requirements – performance and usability.

1 Introduction

Use cases form a standard mechanism of capturing software system requirements. Use cases of an enterprise application are documented using UML Use case diagrams. Use case diagrams are further detailed to include scenarios of system user interaction, and conditions for execution. Documenting use cases has been detailed in [1, 2]. Narrative Text has been recommended in [3] to ensure simplicity and active participation of all the stakeholders. However, in practice simplicity and natural language causes ambiguity and multiple interpretations. Verification of use cases by the stakeholders is a challenge leading to several change requests raised through the later stages of system development. This results in additional development effort and causes deployment delays.

Enterprises comprise of applications that automate a set of business processes. Hence, use of business process models to derive use cases for these applications is recommended and practiced. Business processes are elicited using process maps or workflows. BPMN (Business Process Modeling Notation) [4], an outcome of the standards body - Business Process Management Initiative (BPMI), is one such work-

flow language used by enterprises for documenting their processes. BPMN is the graphical notation for representing business process and most of its notations cater to the business users who are comfortable with flow charts representing process flows. The paper proposes a Use case Specification Framework that uses of BPMN to represent details of the system requirements or to document use cases. The framework aims at enabling business users to understand and verify the functionality of the system in a specification they are most familiar with.

UML [5] proposes use of sequence or activity diagrams to document use cases. However, activity diagrams provide an illustration of what happens in a workflow and have notations that are a subset of business process workflow notations. Hence, the paper proposes a framework which helps in eliciting use cases by incorporating Business Process Modeling to document the scenarios of use cases. The major contribution of this framework is

- Detail use case scenarios using process flow notations that enables business users to understand and verify the application functionality
- Capture additional inputs along with the process flow to enable capture of non functional requirements vis-à-vis usability and performance requirements.

The rest of the paper is organized as follows: Section 2 briefly reviews the existing techniques. Section 3 details the principles used for specifying functional and non-functional requirements using Use case Specification Framework. Section 4 concludes the paper with contributions and future research possible.

2 Related Work

Researchers and practitioners have spent significant effort to improve the clarity of use cases. Use case specification using formal semantics like Abstract State Machine Language (AsML) [6] enables verification and analysis of the use case scenarios or flow of events. Similar is the advantage of representing the use case using Finite state machines and State Charts as described in [7, 8]. Finite state machines (FSM) are helpful in detecting missing events and integration issues. Use case maps [9] are used in describing the use case scenarios using abstract application components. They are used to realize the use cases during the analysis and design phase of the application development where the application components are identified.

All the above techniques are useful to the software development team during analysis, after use cases that have been documented, verified and commonly understood by business users and the development team. UML proposes use of activity diagrams or sequence diagrams to details the use cases. While these diagrams handle most of the scenario representations, it has been observed that BPMN is more suitable of representing complex scenarios like Inclusive OR of two parallel operations, looping of operations. Hence, we propose the use of business workflow and BPMN to detail the use case. The advantage of using BPMN is

- Business users are familiar with business process and workflow models.
- BPMN has notations to describe complex scenarios – Parallel splits and joins Loops and events.