Empirical Evaluation of Test Driven Modeling

Stefan Zugal, Department of Computer Science, University of Innsbruck, Innsbruck, Austria
Cornelia Haisjakkil, Department of Computer Science, University of Innsbruck, Innsbruck, Austria
Jakob Pinggera, Department of Computer Science, University of Innsbruck, Innsbruck, Austria
Barbara Weber, Department of Computer Science, University of Innsbruck, Innsbruck, Austria

ABSTRACT

Declarative approaches to process modeling are regarded well suited for highly volatile environments as they provide a high degree of flexibility. However, problems in understanding and maintaining declarative process models impede their usage. To compensate for these shortcomings, Test Driven Modeling (TDM) has been proposed. This paper reports on an empirical investigation in which TDM is viewed from two different angles. First, the impact of TDM on communication is explored in a case study. Results indicate that domain experts are inclined to use test cases for communicating with the model builder (system analyst) and prefer them over the process model. The second part of the investigation, a controlled experiment, investigates the impact of TDM on process model maintenance. Data gathered in this experiment indicates that the adoption of test cases significantly lowers cognitive load and increases the perceived quality of changes.

Keywords: Business Process Management, Case Study, Controlled Experiment, Declarative Business Process Model, Empirical Research, Test Driven Modeling

INTRODUCTION

In today’s dynamic business environment, the economic success of an enterprise depends on its ability to react to various changes like shifts in customer’s attitudes or the introduction of new regulations (Lenz & Reichert, 2007). Process-Aware Information Systems (PAISs) offer a promising perspective on shaping this capability, resulting in growing interest to align information systems in a process-oriented way (Dumas, van der Aalst, & ter Hofstede, 2005). Yet, a critical success factor in applying PAISs is the possibility of flexibly dealing with process changes (Lenz & Reichert, 2007). To address the need for flexible PAISs, competing paradigms enabling process changes and process flexibility have been developed, e.g., adaptive processes (Reichert & Dadam, 1998), case handling (van der Aalst, 2005) or declarative processes (Pesic, Schonenberg, Sidorova, & van der Aalst, 2007); an overview is provided in (Weber, Reichert, & Rinderle, 2008).
Especially declarative processes have recently attracted the interest of researchers, as they provide a high degree of flexibility (Weber, Reichert, & Rinderle, 2008). Nevertheless, declarative processes are not widely adopted in practice yet. In particular, as pointed out in (Pesic, 2008; Zugal, Pinggera, & Weber, 2012), understandability problems and maintainability problems hamper the usage of declarative process models (for a general discussion about model understandability, we refer to (Zugal, Pinggera, & Weber, 2011a; Zugal, Pinggera, Weber, Mendling, & Reijers, 2011)). An approach tackling these problems, the Test Driven Modeling (TDM) methodology, is presented in Zugal, Pinggera, and Weber (2012). TDM aims at improving the understandability and maintainability of declarative process models as well as the communication between domain expert and model builder (system analyst) by adopting the concept of test cases from software engineering. While the proposed concepts seem to be beneficial from a theoretical point of view, no comprehensive empirical evaluation exists yet. The goal of this paper is to extend previous work (Zugal, Pinggera, & Weber, 2011c) and provide a more comprehensive empirical evaluation. Hence, in this paper, we will provide two complementary perspectives. First, we will embrace a qualitative angle and report the results of a case study. Therein, we focus on the question in how far communication between the domain expert and the model builder is influenced by the adoption of TDM. Second, we will look into the impact of TDM on the maintainability of declarative process models in a controlled experiment, i.e., adopting a quantitative angle. The contribution of this paper is twofold. On the one hand, we report from a case study that investigates the impact of TDM on communication. On the other hand, results from a controlled experiment are integrated to provide a more comprehensive picture.

The remainder of this paper is structured as follows. First, background information about TDM is provided. Then, results from a case study and a controlled experiment are reported. Insights and limitations are subsequently discussed before the paper is closed with the conclusion.

BACKGROUND

In this section background information is provided. First, declarative processes and associated problems are discussed. Then, it is sketched how TDM aims at resolving these problems.

Declarative Processes

There has been a long tradition of modeling business processes in an imperative way. Process modeling languages supporting this paradigm, like BPMN, EPC and UML Activity Diagrams, are widely used. Recently, declarative approaches have received increasing interest and suggest a fundamentally different way of describing business processes (Pesic, 2008). While imperative models specify exactly how things have to be done, declarative approaches only focus on the logic that governs the interplay of actions in the process by describing the activities that can be performed, as well as constraints prohibiting undesired behavior. An example of a constraint in an aviation process would be that crew duty times cannot exceed a predefined threshold. Constraints described in literature can be classified as execution and termination constraints (Zugal, Pinggera, & Weber, 2012). Execution constraints, on the one hand, restrict the execution of activities, e.g., an activity can be executed at most once. Termination constraints, on the other hand, affect the termination of process instances and specify when process termination is possible. For instance, an activity must be executed at least once before the process can be terminated. Most constraints focus either on execution or termination semantics. However, some constraints also combine execution and termination semantics (e.g., the succession constraint (Pesic, 2008)).

To illustrate the concept of declarative processes, a declarative process model is shown in Figure 1 a). It contains activities $A$ to $F$ as
A Tool Support for Secure Software Integration
[www.igi-global.com/article/tool-support-secure-software-integration/43925?camid=4v1a](www.igi-global.com/article/tool-support-secure-software-integration/43925?camid=4v1a)

Tourists' Mobile Information Seeking Behavior: An Investigation on China's Youth
[www.igi-global.com/article/tourists-mobile-information-seeking-behavior/188595?camid=4v1a](www.igi-global.com/article/tourists-mobile-information-seeking-behavior/188595?camid=4v1a)