

Handling Events During Business Process Execution: An Empirical Test

Barbara Weber¹, Jakob Pinggera¹, Stefan Zugal¹, and Werner Wild²

¹Quality Engineering Research Group, University of Innsbruck, Austria
{Barbara.Weber,Jakob.Pinggera,Stefan.Zugal}@uibk.ac.at,

²Evolution Consulting, Innsbruck, Austria
Werner.Wild@evolution.at

Abstract. Declarative approaches have been proposed to counter the limited flexibility of the imperative modeling paradigm, but little empirical insights are available into their actual strengths and use. Our previous work has shown that end-users can effectively model and execute a declarative process with a considerable spectrum of constraints. However, what is still unclear is how effectively end-users are able to handle unforeseen events that can occur during run-time. This paper describes the design, execution, and results of a controlled experiment in which subjects have to execute a process with varying levels of events. The results suggest that our subjects, while being able to effectively handle constraints, have difficulties to handle unforeseen events during run-time. This outcome supports the argument that declarative processes require more experienced people, especially when dealing with unforeseen events.

1 Introduction

In today's dynamic business environment the economic success of an enterprise depends on its ability to react to change, like shifts in customers' attitudes or the introduction of new laws [1]. Process-aware information systems (PAISs) offer a promising perspective on shaping this capability, resulting in a growing interest to align information systems in a process-oriented way [2]. Yet, a critical success factor when applying a PAIS is the option to flexibly deal with process changes [3]. To address the need for flexible PAISs, competing paradigms enabling process changes and process flexibility have been developed, e.g., adaptive processes [4], case handling [5], declarative processes [6], and late binding and modeling [7] – for an overview see [8]. All of these approaches relax the strict separation of build-time (i.e., modeling or planning) and run-time (i.e., execution), which is typical for traditional workflow management systems following the imperative paradigm. However, by closely interweaving planning and execution the above mentioned approaches allow for a more agile way of planning. In particular, users are empowered to defer decisions regarding the exact control-flow to run-time, when more information is available. Depending on the concrete approach, planning and execution are interwoven to different degrees, resulting in different levels of decision deferral. The highest degree of decision deferral is enabled

by declarative processes, which describe activities that can be performed as well as constraints preventing undesired behavior [8]. A declarative approach, therefore, seems to be particularly promising for highly dynamic processes [6, 9]. The support for partial workflows [9] allowing users to defer decisions to runtime [8], the absence of over-specification [6], and more maneuvering room for end-users [6] are all advantages commonly attributed to declarative processes.

Although the benefits of declarative approaches seem rather evident, such approaches are not yet widely adopted in practice. In addition, there is a lack of empirical evidence on how well declarative approaches perform in real-world settings. In our previous work we have shown that end-users can effectively model and execute declarative processes even with a considerable spectrum of constraints, especially when appropriate tool support is in place [10]. However, it is still unclear how well end-users can handle unforeseen events during the execution of declarative processes.

The goal of this paper is to pick up on the demand for more empirical insights into the use of declarative approaches. Specifically, we aim to investigate how the occurrence of exceptional situations may impede end-users' success when using a declarative approach for handling a particular business case (i.e., process instance). Proponents of declarative approaches argue that they are especially suited to support dynamic processes and that handling of unforeseen events is one of the strengths of the declarative approach [6, 9]. Due to its high flexibility the declarative approach provides maneuvering room for end-users to react upon unforeseen events without necessarily having to deviate from the process model. However, following literature on agile methods one could also argue that talent and skills are among the critical people-factors [11, 12] and declarative processes tend to necessitate a richer mix of higher-skilled people than traditional imperative approaches.

This paper reports on the results of a controlled experiment investigating how well inexperienced users can handle unforeseen events during the execution of declarative processes. Its findings are based on an experiment conducted in December 2008 at the Management Center Innsbruck with 20 students. The structure of this paper is as follows. After providing necessary background information in Section 2, Section 3 describes the experimental definition and Section 4 deals with the execution of the experiment and presents the results. Related work is listed in Section 5, Section 6 concludes the paper with a summary and an outlook.

2 Background

This section introduces declarative processes as well as the software used for the experiment, the Alaska Simulator.

2.1 Declarative Processes

There is a long tradition of modeling business processes in an *imperative* way. Process modeling languages supporting this paradigm, like BPMN, BPEL and

UML Activity Diagrams, are widely used. Recently, *declarative* approaches have received increased interest and suggest a fundamentally different way of describing business processes [13]. 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 (1) the activities that can be performed, as well as (2) constraints prohibiting undesired behavior. An example of a constraint in a travel process would be that between a **Diving** activity and a **Flightseeing** activity there must be a resting period of two days to prevent aeroembolism. Imperative models take an ‘inside-out’ approach by requiring all execution alternatives to be explicitly specified in the model. Declarative models, in turn, take an ‘outside-in’ approach: constraints implicitly specify execution alternatives, as all valid alternatives have to satisfy the constraints [14]. Adding more constraints means discarding some execution alternatives (cf. Fig. 1). This results in a coarse up-front specification of a process, which can then be refined iteratively during run-time. Typical constraints described in literature can be roughly divided into three classes (e.g., [7, 13]): constraints restricting the *selection* of activities (e.g., the minimum or maximum occurrence of activities, mutual exclusion, co-requisite), the *ordering* of activities (e.g., pre-requisite or response constraints) and the use of *resources* (e.g., execution time of activities, time difference between activities, budget, etc.).

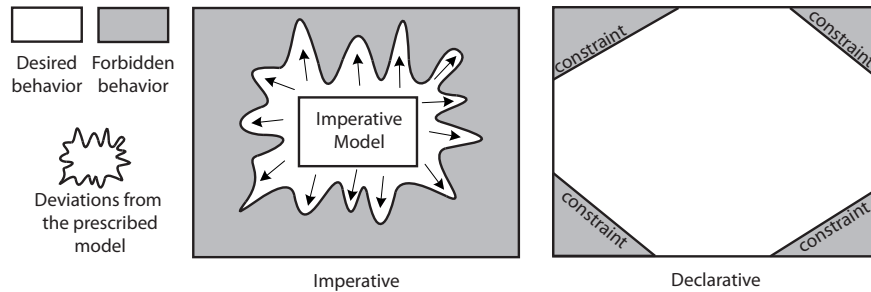


Fig. 1. Imperative vs. Declarative Approaches to Process Modeling (adapted from [14])

2.2 The Alaska Simulator

The Alaska Simulator¹ [15] fosters the comparison of different approaches to process flexibility, e.g., declarative processes, by using a *journey* as metaphor for a *business process*. The similarities being exploited here are that regardless of whether a journey or a business process is executed, various steps must be planned and carried out, even if the actual execution of those steps may be different from what is initially foreseen. Furthermore, journey planning is an attractive context for many people to become engaged in, which highly improves their willingness to participate in experiments.

The actions of a journey, like travel activities, routes and overnight stays correspond to activities in the business process. When conducting a journey, the

¹ Developed at the University of Innsbruck, <http://www.alaskasimulator.org>

goal is to maximize the travel experience (i.e., the overall “business value” of the journey), typical goals for business processes are the minimization of cost, cycle time or the optimization of quality or customer satisfaction. For optimizing the execution of a particular business case, information about the benefits (i.e., business value), cost and duration of activities is essential. Furthermore, both journeys and highly flexible business processes are characterized by incomplete information prior to execution. The business value for executing a particular activity within a business process is usually uncertain, likewise the outcome of a travel activity is not predefined and varies with the *weather conditions* encountered. The degree of variation is defined by the activity’s *reliability*, i.e., low reliability indicates that the outcome of the activity is highly weather dependent. The overall business value of a journey (i.e., a numeric value representing the travel experience) is calculated as the sum of business values of all performed activities. Prior to performing the journey only the expected business value for each activity as well as its reliability (see below) are known. During the journey the activity’s actual business value is calculated based on the weather conditions encountered. In addition to changing weather conditions, *unforeseen events* (e.g., a traffic jam resulting in delays) create uncertainty in a journey, while changing requirements or new laws complicate the modeling and execution of business processes. When composing a concrete business case, different constraints like *selection constraints*, *ordering constraints* or *resource constraints* have to be considered (cf. Section 2.1), similar constraints also exist when planning a journey (e.g., mandatory activities, dependencies between activities). To assess the last responsible moment for committing to an action, users must consider both its *availability* and *reliability*. By firmly booking an action its availability can be guaranteed, but the cost of the action must be paid immediately. If the booking is canceled during the journey, a cancellation penalty might apply, thus making too early commitments costly. Furthermore, booking is only possible up to a certain time before executing the action, as specified by the booking deadline.

Fig. 2 depicts the graphical user interface of the Alaska Simulator. Users can compose their individual travel plan by dragging available actions from the Available Actions View (3) onto the Itinerary (1). Actions are only available at a particular location on the Map (4). Existing constraints are displayed in the Constraint Overview (2) and have to be considered when composing a concrete journey. After each user (inter-)action, the journey is validated and the user is informed about any constraint violations and inconsistencies in the plan (5).

3 Experimental Definition and Planning

The main goal of the experiment is to evaluate the effects of unforeseen events on process outcomes (i.e., business value and number of failed journeys). Section 3.1 describes the setup of the experiment, the design is elaborated in Section 3.2. Finally, Section 3.3 discusses possible risks threatening the validity of the experiment as well as countermeasures taken.

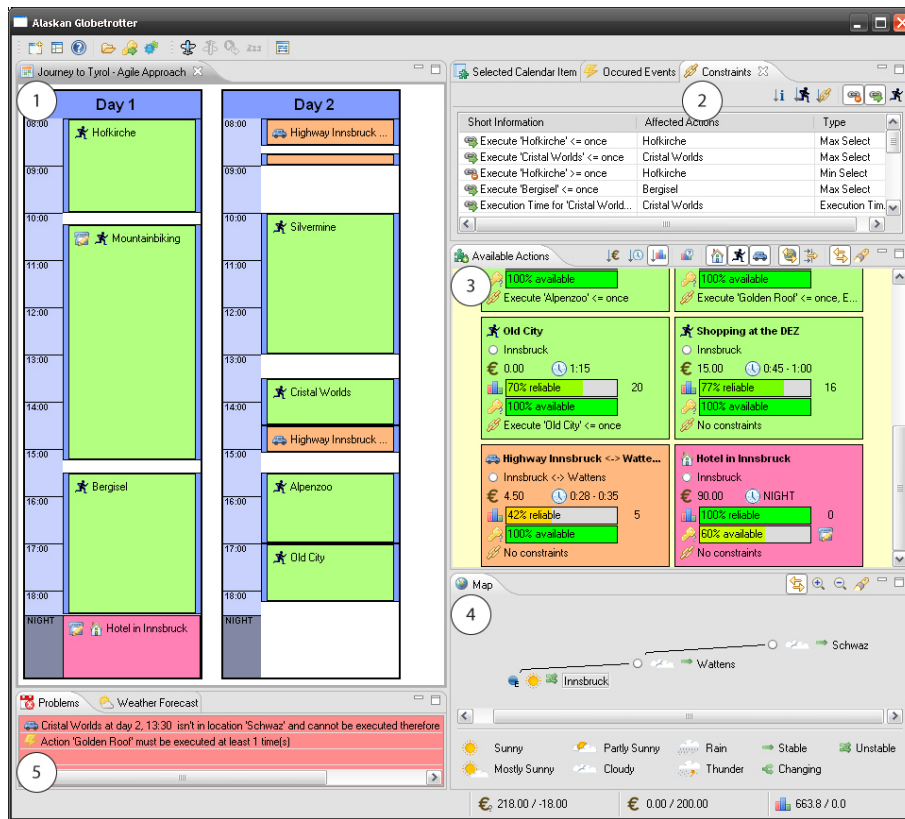


Fig. 2. Screenshot of the Alaska Simulator

3.1 Experimental Setup

Subjects: We conducted the experiment with 20 students of the study program “Management, Communication & IT” at the Management Center Innsbruck.

Objects: Two journeys representing two different business processes are used as objects, subsequently referred to as Configuration California (ConfCA) and Configuration Alaska (ConfAK). The configurations define the journey settings, like actions to be executed, constraints restricting their execution, events that might occur during run-time and weather conditions (cf. Section 2). For each of the configurations, two variants are created: A and B, differing in the number of events only. While Variant A contains no events, Variant B comprises many unforeseen events (e.g., event increasing the action’s duration, temporary road closure). An overview of the different variant characteristics is given in Fig. 3.

Factor and Factor Levels: The number of unforeseen events that occur during run-time is the considered factor with levels “no events” and “many events”. Variant A of a configuration corresponds to factor level “no events” and variant B to factor level “many events”.

Variant	Events	Constraints
Alaska A		One budget constraint, one end-location constraint, three execution time constraints, three mandatory actions, one constraint requiring A to be followed by n times B and a final C, one pre-requisite constraint, one constraint requiring a minimum delay between two actions, one mutual exclusion constraint
Alaska B	Three events increasing the action's duration, two events increasing the action's duration and business value	
California A		One budget constraint, one end-location constraint, three mandatory actions, two execution time constraints, two constraint requiring a minimum delay between two actions
California B	Three events increasing the action's duration, one event increasing the action's duration and business value, one event closing a road for a certain period of time	

Fig. 3. Characteristics of the Configuration Variants

Response Variable: The achieved *business value* when planning and executing a given configuration with a given level of events is the response variable (cf. Section 2 for a description on how the business value is calculated). In addition, the *number of failed journeys*, i.e., journeys which could not be completed without constraint violations, is considered. To ensure comparability of results, weather conditions are the same for each subject.

Hypothesis Formulation: Goal of the experiment is to investigate the impact of unforeseen events on the response variables *business value* and *number of failed journeys*. Accordingly, we postulate the following hypotheses:

- **Null Hypothesis $H_{0,0}$:** There is no significant difference in the mean business values between configurations irrespective of the number of events.
- **Null Hypothesis $H_{1,0}$:** There is no significant difference in the number of failed journeys between configurations irrespective of the number of events.

Instrumentation: To ensure precise measurement of business value, the Alaska Simulator provides a mechanism for logging each relevant step the user undertakes while planning and executing a journey.

3.2 Experimental Design

The experimental setup is based on the guidelines for designing experiments in [16]. Following these guidelines a *randomized balanced single factor* experiment is conducted with *repeated measurements*. The experiment is called *randomized*, since subjects are assigned to groups randomly. We denote the experiment as *balanced* as each factor level is used by each subject, i.e., each student plans and executes two journeys, one without and one with events. As only a single factor is manipulated (i.e., the level of events), the design is called *single factor*. Due to the balanced nature of the experiment, each subject generates data for both factor levels and thus provides *repeated measurements*. Fig. 4 depicts the design following the aforementioned criteria. The subjects are randomly assigned to two groups of equal size, subsequently referred to as Group 1 and Group 2. To provide a balanced experiment with repeated measurements, the overall procedure consists of two runs. In the first run Group 1 applies factor level *no events* to object ConfCA, Group 2 factor level *many events* to the same object. In the second run, factor levels are switched and Group 1 applies factor level *many events* on ConfAK, Group 2 factor level *no events* to the same object. Since no subject deals with an object more than once, this design avoids learning effects.

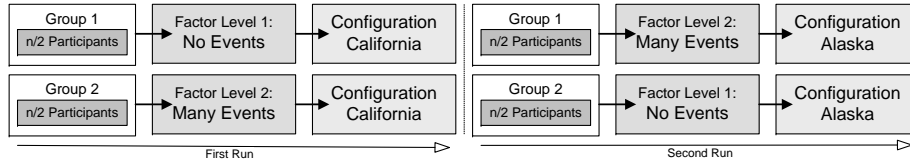


Fig. 4. Employed Experimental Design

3.3 Risk Analysis and Mitigation

In this section risks threatening the validity of the experiment are discussed.

Individual Planning Experience: Differences of participating students in respect to planning experience and productivity might have an impact on the students' performance, i.e., the business value achieved. This issue can be balanced by conducting the experiment with a sufficiently large and representative set of students or by replicating the experiment. The relatively low number of subjects (19 out of 20 could be used for data analysis) certainly constitutes a threat to the validity of this experiment.

Suitability of Metaphor: Whether or not the results of this experiment can be generalized to business process modeling and execution highly depends on the suitability of the chosen metaphor. However, due to the similarities of business process modeling and traveling planning and their respective execution (cf. Section 2), we assume the suitability of the metaphor. To further increase confidence in our view we plan testing whether travel planning serves as a good proxy for business process modeling and execution in future experiments.

Students instead of Professionals: In our experiment undergraduate students with limited planning experience were the subjects for investigating how well inexperienced users are able to model and execute declarative processes with varying numbers of events. While students can be regarded as suitable proxies for inexperienced users [17], it is arguable whether the results of this experiment can be generalized to professionals with significant planning experience. When replicating the experiment with experienced professionals we expect them to clearly obtain better process outcomes (i.e., higher business value and lower number of failed journeys).

Choice of Object: To be able to generalize results gained from this experiment, the configurations must be representative for a wide range of business process settings. Although the configurations used in this experiment do not have the complexity of real-world processes, they range well beyond the size of toy examples and include 22 and 26 activities, each of them varying in terms of expected business value, reliability and availability as well as their constraints and events.

Team Planning: In our experiment planning and execution was done on an individual basis, not in teams. Since planning often involves interactions among domain experts, system analysts and stakeholders, it has to be investigated how far our results can be transferred to team planning. For this we plan to replicate the experiment in a team setting.

4 Performing the Experiment

Section 4.1 describes the experiment’s preparation and execution. Then, Section 4.2 presents the analysis of data for our experiment, followed by a discussion of the results in Section 4.3.

4.1 Experimental Operation

Experimental Preparation: The preparation of the experiment included the elaboration of the experimental design, implementation of the Alaska Simulator and devising the two travel configurations, i.e., ConfCA and ConfAK. To ensure that each configuration is correct and can be executed in the available amount of time, we performed pre-tests with several persons of different backgrounds.

Experimental Execution: The experiment was conducted in December 2008 at the Management Center Innsbruck. For organizational reasons, the execution of the experiment was split into two distinct sessions with 10 students each. At the beginning of each session, an introductory lecture was given to familiarize everyone with the Alaska Simulator and to clarify the experiment’s rules and goals. For this, the students received a “starter kit” consisting of screencasts explaining the main features of the Alaska Simulator. Having watched the screencasts, the students were then randomly assigned to one of the two groups. As pointed out in Section 3.2, the experiment was executed in two subsequent runs, each taking about one hour. During the first 25 minutes of each run students could explore the configuration (i.e., ConfCA for the first run, ConfAK for the second run) to gather relevant domain knowledge. In the remaining 35 minutes students had to plan and execute the journey with the goal of optimizing the business value.

Data Validation: After having conducted the experiment, logged data was analyzed. We discarded data from one student since he did not follow the experiment setup (i.e., he adopted the same planning approach twice). Thus, 19 subjects remained for data analysis.

4.2 Data Analysis

In the following we describe the analysis and interpretation of data.

Descriptive Analysis: Based on data obtained from the logs of the Alaska Simulator, descriptive statistics for response variables *business value* and *number of failed journeys* were calculated.

Configuration	Approach	N	Failed Journeys	MinBV	MaxBV	MeanBV	Standard Deviation BV
California	No Events	9	0	5925	7492	6581	573
California	Many Events	10	4	4815	7289	5883	693
Alaska	No Events	10	1	2895	5497	4114	827
Alaska	Many Events	9	6	1045	4907	2826	1179

Fig. 5. Descriptive Statistics for Response Variables

Fig. 5 shows that for both ConfCA and ConfAK, Variant A (no events) yields a higher mean business value and a lower number of failed journeys compared

to Variant B (many events). The question is whether the observed differences in mean business values and number of failed journeys are statistically significant. **Data Plausibility:** Fig. 6 shows a *box-whisker-plot diagram* as used for analyzing data plausibility. It visualizes data distribution and detects outliers. For ConfAK (many events) a single outlier exists. Since this is the only outlier, plausible data distributions seem to be in effect.

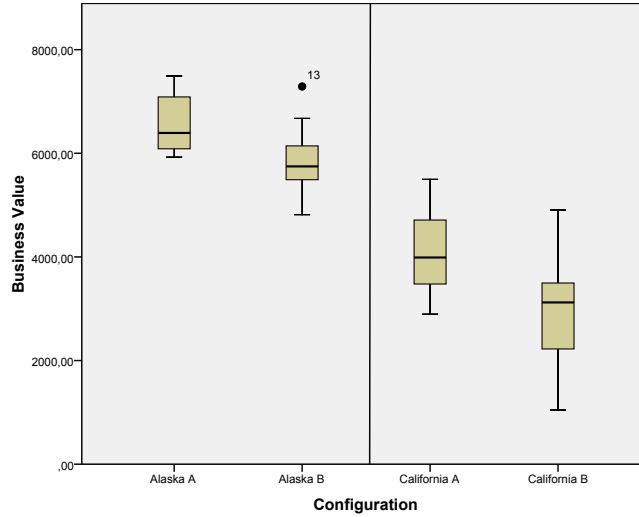


Fig. 6. Data Distribution (Box-Whisker-Plot Diagram)

Testing for Differences in Business Value: Since the expected business values of ConfAK and ConfCA differ, hypothesis testing is performed for each configuration separately. For ConfCA preconditions for the t-test for homogeneous variances are fulfilled (i.e., data is normally distributed and the Levene test confirmed equal variances). With an obtained significance of 0.013 (< 0.05) hypothesis $H_{0,0}$ can be rejected at a confidence level of 95%. ConfAK also fulfills all prerequisites for the t-test for homogeneous variances. The resulting significance of 0.030 (< 0.05) also leads to a rejection of hypothesis $H_{0,0}$ at a confidence level of 95%.

Testing for Differences in Number of Failed Journeys: To test for differences in the number of failed journeys we used Fisher's exact test [18]; the Chi Square test was not applicable due to the small sample size. For ConfCA, with an obtained significance of 0.017 (< 0.05) hypothesis $H_{1,0}$ can be rejected at a confidence level of 95%. For ConfAK, in turn, the obtained p-value of 0.054 (> 0.05) is slightly above the cut-off point. Consequently, hypothesis $H_{1,0}$ cannot be rejected for ConfAK.

4.3 Discussion of Results

The major finding from our data analysis is that unforeseen events have a statistically significant impact on the outcome of journeys. Furthermore, the results

obtained in this experiment seem to confirm the findings reported in [10], suggesting that handling constraints causes little difficulties for end-users, especially if appropriate tool support is provided. The constraints used in our experiment show a similar level of complexity as those used in [10]. As only one out of 19 journeys without unforeseen events was not completed successfully, we conclude that only when combined with events, constraints have a significant impact on the business value of a journey.

A manual analysis of our data showed that some events were more critical for the the process outcome (i.e., business value and number of failed journeys) than others. Especially, events that occurred in combination with mandatory actions were causing difficulties. For example, one of the events our subjects had to handle was a traffic jam on a route to a location with a single mandatory activity. This mandatory activity had a low availability and a restricted execution time. To handle this event, our subjects could not simply move the action, but usually had to rearrange major parts of the journey to fulfill the constraints and use the remaining time as effectively as possible.

Our results also support the argument that talent and skills are among the critical people-factors for agile methods as enabled by declarative processes [11, 12]. In fact, only few students were able to effectively handle events and to fully exploit the flexibility provided by the declarative approach. As stated in [10], tool support was essential for the successful dealing with constraints. Analogously, we assume that tools supporting the user's decision making, e.g., recommendation systems [19], might be beneficial for the journey's outcome.

5 Related Work

Most existing work about flexibly dealing with exceptions, changes, and uncertainty in the context of PAISs and related technologies is strongly *design-centered*, i.e., aiming at the development of tools, techniques, and methodologies. For overviews and discussions of these approaches, see [8, 20, 21].

Only few empirical investigations exist that aim to establish the suitability of the various proposed artifacts. Closely related to this paper is our previous work, which investigates how well end-users can cope with the gained flexibility provided by declarative approaches, especially when processes become rather complex [10]. While [10] focuses on the impact of constraints, this paper investigates the impact of unforeseen events. Also closely related is our work on the comparison of agile and plan-driven approaches to business process modeling and execution [22]. A theoretical discussion on declarative versus imperative approaches is provided in [23]. In [24], the results of a controlled experiment comparing a traditional workflow management system and case-handling are described. The systems are compared with respect to their associated implementation and maintenance efforts. In turn, the impact of workflow technology on PAIS development and PAIS maintenance is investigated in [25]. However, these works primarily focus on traditional workflow technology, while this paper puts its emphasize on declarative approaches. Other empirical works with

respect to PAISs mainly deal with establishing their contribution to business performance improvement, e.g. [26, 27], and the way end-users appreciate such technologies, e.g., [28, 29].

6 Summary and Outlook

The advantages attributed to declarative processes are manifold, e.g., support for partial workflows allowing users to defer decisions to run-time, the absence of over-specification as well as more room for end-users to maneuver. However, their practical application requires the ability to resolve uncertainty and exploit learning outcomes. This raises the question whether inexperienced users are able to execute declarative processes especially when unforeseen events occur during run-time. This work picks up on this demand and contributes a controlled experiment comparing the process outcome for inexperienced users depending on the number of events.

While previous work shows that end-users can effectively handle varying levels of constraints when executing a declarative process, this paper demonstrates that unforeseen events are more problematic. The major result of our experiment is that process outcomes of inexperienced planners are significantly affected by unforeseen events which have to be handled during run-time. In particular, the combination of constraints and events turned out to be challenging for our subjects. These findings support the argument that declarative approaches require experienced users to fully exploit its benefits.

For further research we aim to investigate different techniques for improving understandability and maintainability of declarative process models to facilitate their application by less experienced users. Furthermore, we plan to run experiments in settings where planning is done in small teams, not individually, and replicate the experiment with more experienced users.

References

1. Poppendieck, M., Poppendieck, T.: Implementing Lean Software Development: From Concept to Cash. Addison-Wesley (2006)
2. Weske, M.: Business Process Management: Concepts, Methods, Technology. Springer (2007)
3. Mutschler, B., Reichert, M., Bumiller, J.: Unleashing the effectiveness of process-oriented information systems: Problem analysis, critical success factors and implications. *IEEE Trans. on Systems, Man, and Cybernetics* **38** (2008) 280–291
4. Reichert, M., Dadam, P.: ADEPT_{flex} – Supporting Dynamic Changes of Workflows Without Losing Control. *JHIS* **10** (1998) 93–129
5. Van der Aalst, W., Weske, M., Grünbauer, D.: Case handling: A new paradigm for business process support. *Data and Knowledge Engineering*. **53** (2005) 129–162
6. Pesic, M., Schonenberg, M., Sidorova, N., van der Aalst, W.: Constraint-Based Workflow Models: Change Made Easy. In: Proc. CoopIS’07. (2007) 77–94
7. Sadiq, S., Sadiq, W., Orłowska, M.: A Framework for Constraint Specification and Validation in Flexible Workflows. *Information Systems* **30** (2005) 349 – 378

8. Weber, B., Reichert, M., Rinderle-Ma, S.: Change patterns and change support features -enhancing flexibility in process-aware information systems. *Data and Knowledge Engineering* (2008) 438–466
9. Wainer, J., Bezerra, F., Barthelmeß, P.: Tucupi: a flexible workflow system based on overridable constraints. In: *Proc. SAC '04*. (2004) 498–502
10. Weber, B., Reijers, H.A., Zugal, S., Wild, W.: The declarative approach to business process execution: An empirical test. In: *Proc. CAiSE'09*. (2009) 470–485
11. Highsmith, J., Cockburn, A.: *Agile Software Development: The Business of Innovation*. *IEEE Computer* **34** (2001) 120–122
12. Lindvall, M.: *Empirical Findings in Agile Methods*. In: *Proc. XP/Agile Universe '02*. (2002) 197–207
13. van der Aalst, W., Pesic, M.: *DecSerFlow: Towards a Truly Declarative Service Flow Language*. Technical report, BPMcenter.org (2006)
14. Pesic, M.: *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Eindhoven University of Technology (2008)
15. Weber, B., Zugal, S., Pinggera, J., Wild, W.: *Experiencing Process Flexibility Patterns with Alaska Simulator*. In: *Proc. BPMDemos '09*. (2009)
16. Wohlin, C., Runeson, R., Halst, M., Ohlsson, M., Regnell, B., Wesslen, A.: *Experimentation in Software Engineering: an Introduction*. Kluwer (2000)
17. Runeson, P.: Using students as experiment subjects: An analysis on graduate and freshmen student data. In: *Proc. EASE'03*. (2003) 95–102
18. Fleiss, J.L.: *Statistical Methods for Rates and Proportions*. Wiley (1981)
19. Schonenberg, H., B. Weber, B.F. van Dongen, W.v.d.A.: Supporting flexible processes through recommendations based on history. In: *Proc. BPM'08*. (51-66) 2008
20. Kammer, P., Bolcer, G., Taylor, R., Hitomi, A., Bergman, M.: *Techniques for Supporting Dynamic and Adaptive Workflow*. *Computer Supported Cooperative Work* **9** (2000) 269–292
21. Reijers, H., Rigter, J., van der Aalst, W.: The case handling case. *International Journal of Cooperative Information Systems*. **12** (2003) 365–391
22. Weber, B., Reijers, H.A., Zugal, S., Wild, W.: The Declarative Approach to Business Process Execution: An Empirical Test. In: *Proc. CAiSE '09*. (2009) 270–285
23. Fahland, D., Mendling, J., Reijers, H.A., Weber, B., Weidlich, M., Zugal, S.: Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: *Proc. EMMSAD '09*. (2009) 353–366
24. Mutschler, B., Weber, B., Reichert, M.: Workflow management versus case handling - results from a controlled software experiment. In: *Proc. SAC'08*. (2008) 82–89
25. Kleiner, N.: Supporting usage-centered workflow design: Why and how?. In: *Proc. BPM'04*. (2004) 227–243
26. Oba, M., Onoda, S., Komoda, N.: Evaluating the quantitative effects of workflow systems based on real case. In: *Proc. HICSS'00*. (2000)
27. Reijers, H., van der Aalst, W.: The effectiveness of workflow management systems: Predictions and lessons learned. *International Journal of Information Management* **25** (2005) 458–472
28. Bowers, J., Button, G., Sharrock, W.: Workflow from within and without: technology and cooperative work on the print industry shopfloor. In: *Proc. CSCW'95*. (1995) 51–66
29. Poelmans, S.: Workarounds and distributed viscosity in a workflow system: a case study. *ACM SIGGROUP Bulletin* **20** (1999) 11–12