Declarative Versus Imperative Business Process Languages
A Controlled Experiment

Natália C. Silva, César A. L. de Oliveira, Fabiane A. L. A. Albino and Ricardo M. F. Lima
Center for Informatics (CIn), Federal University of Pernambuco, Recife, PE, Brazil

Keywords: Flexible Business Processes, Controlled Software Experiment, Comparative Study, Declarative Languages.

Abstract: It has been argued that traditional workflows lack the flexibility to cope with complex and changing environments found in several business domains. The declarative approach surged with the aim of enabling more flexible business process management systems. Processes are designed in terms of activities and rules that constrain their execution. As such, declarative models are less rigid and prescriptive than workflows, since this approach focus on modeling *what* must be done and not *how*. Despite these arguments, there is no quantitative evidence that the benefits provided by current declarative approaches outperform the features of traditional workflows. In this work, we present the results of a controlled experiment conducted to empirically compare Workflow and Declarative approaches to business process modeling. A set of test cases is proposed to the participants to test their capacity to cope with unexpected situations. Our findings do not show signficative differences from adopting one process modeling approach or the other.

1 INTRODUCTION

Workflow technology is a popular approach for the automation of business processes (Jeston and Nelis, 2006) (Coalition, 1995). Workflows describe each activity that must be performed during the execution of a business process, the order in which they need to be executed, the participants that execute each one, and the flow of information between them (zur Muehlen, 2002). The Workflow approach is widely used by business process practitioners. Business process automation provides higher efficiency, less coordination effort, and higher quality to business processes (H. Reijers and van der Aalst, 2003).

However, the Workflow approach assumes that the process modeler knows in advance every step that will be necessary during the process execution for producing the specified output. Due to this prescriptive nature, many researchers argue that Workflow models suffer from excessive complexity and that this approach undermine the capacity of companies to deal with unforeseen situations (Pesic, 2008)(Weber et al., 2009)(Dadam et al., 2008)(Nurcan, 2008). As the environment where the business processes execute increases in complexity, the Workflow models become less effective in handling the incoming cases. They often are either too simple, thus unable to handle the variety of situations that occur, or too complex, trying to model every imagined possible situation but being hard to maintain. In both cases they may cause several problems to the company.

Declarative business processes surged from the necessity for supporting the execution of processes in complex or changing environments (Pesic, 2008). The declarative approach uses descriptive modeling. Business processes are described by means of business rules that tell what one can, can not, or should do to produce the desired output. It tells *what* has to be done, but not *how*. The Declarative approach promises to be able to support a large variety of situations with models that are small and concise (Pesic, 2008).

Despite such arguments in favour of Declarative models over Workflows, no controlled experiment has been presented to empirically demonstrate their validity. In this paper, our interest is to compare these two different process modeling approaches – Workflow and Declarative – in real-world business domains. To accomplish this, a formal experiment was conducted to quantitatively measure the benefits of each approach in face of unstable environments. In this experiment, a group of students had the task of modeling a set of requirements of a real software development company. We measured the effort required by them to model the process and to adapt it in face of unexpected situations. Contrary to the claims that

C. Silva N., A. L. de Oliveira C., A. L. A. Albino F. and M. F. Lima R..
Declarative Versus Imperative Business Process Languages - A Controlled Experiment.
DOI: 10.5220/0004896203940401
In *Proceedings of the 16th International Conference on Enterprise Information Systems* (ICEIS-2014), pages 394-401
ISBN: 978-989-758-029-1

declarative models are better to cope with these situations, the results showed no statistically relevant indications that declarative models can outperform workflow when dealing with unexpected situations.

This paper is structured as follows: Section 2 provides an overview of workflow and declarative approaches and the arguments for moving from workflow models to declarative models. Section 3 details the experiment settings and methodology employed. The execution of the experiment is then described in Sec. 4. Section 5 presents the analysis of the results obtained. Section 6 discuss related work and the final conclusion is summarized in Sec. 7.

2 BACKGROUND

This section gives an overview on workflow and declarative approaches and describes the modeling tools employed in the experiment performed.

2.1 Workflow and Declarative Processes

Workflow Management System (WfMS) technology has grown in use in the last decades, and has become a standard technology for the automation and management of business processes (Coalition, 1995). A workflow describes an algorithm for the execution of a business process. As such, it requires that the modeler knows in advance every step necessary for producing the desired results, for every possible situation. There are a number of modeling languages currently in use for business process modeling and automation through WfMS. Business Process Modeling Notation (BPMN) (White, 2006), Business Process Execution Language (BPEL), and XML Process Definition Language (XPDL) (WFMC, 2002) are prominent examples. Several industrial tools are available for workflow modeling and execution. In this work, we employed the BizAgi free process modeler (Modeler, 2010), which supports Business Process Modeling Notation (BPMN) as the modeling language.

Workflow Management Systems are considered to be too rigid because they do not provide ways to cope with unforeseen situations. Unexpected events must be handled outside of the system (Pesic and van der Aalst, 2006). As a response to the demand for more *flexible* management systems, a new generation of *adaptive* business process management systems has been proposed (Nurcan, 2008)(Dadam et al., 2008). Examples of this kind of systems are Case-Handling Systems (CHS) (Mutschler et al., 2008), Declarative Business Processes (Pesic, 2008), and systems supporting ad-hoc process change (the ADEPT2 framework) (Dadam et al., 2008).

The focus of this work is on the Declarative approach, which proposes the use of *declarative languages* to define the business rules that drive the execution of a business process (Pesic and van der Aalst, 2006). This represents a paradigm shift from traditional imperative languages that are employed by other approaches. The key feature of declarative languages is that they specify "what" must be done but do not determine "how". By such means, they allow the participants of the process to make context-aware decisions during the business process execution. On the other side, constraining business rules are defined to prevent participants from performing activities in a way that is prohibited or undesired by the company.

The DECLARE framework (Pesic, 2008) is a tool for modeling and executing declarative business process models. DECLARE employs rule templates that can be used in the construction of business rules. These templates are mapped into formal expressions described in Linear Temporal Logic (LTL) and interpreted through a reasoning engine to decide which activities are required and which are prohibited during the process execution. A graphical template language, called *ConDec* (Pesic and van der Aalst, 2006), is also available in DECLARE. This language helps in providing a more intuitive, graphical view of the process. It is automatically interpreted as LTL expressions by the DECLARE's execution engine.

ConDec introduces a number of constructions for the definition of the relations between activities in a business process:

- **Existential Templates:** express the number of times an activity can or needs to be executed in the same process instance;
- **Relational Templates:** express dependencies between activities, such as precedence (*B* cannot be executed before *A*), response (*B* needs to be executed after each execution of *A*), and co-existence (if *A* is ever executed, *B* must also be executed in the same instance);
- **Choice Templates:** express decisions. For example, mutual exclusiveness (either *A* or *B* can be executed, but not both);
- **Negation Templates:** a negated version of the relational templates (*B* cannot be executed after *A*).

In this work, we employed an alternative interface based on the DECLARE framework and the ConDec language (Pesic, 2008). Although we intended to use DECLARE as a modeling tool, initial tests with a real business case revealed a number of weaknesses in the tool that prevented us to employ it in the experiment (for example, LTL suffers from state-space explosion when the number of rules grows, so making DECLARE unsuitable). Therefore, we employed another tool, called *Kinetic Process*, that was developed by our research group. This tool provides an interface for the creation and management of business rules based on the same templates of DECLARE, but not relying on LTL for its implementation.

3 EXPERIMENT SCENARIO

The objective of the experiment presented in this paper is to compare workflow and declarative approach in terms of correctness and flexibility of their models. The main questions we want to answer are:

- How easy is to model a declarative process, when compared to traditional workflow?
- Do declarative models successfully cope with unexpected situations where the workflow fail?
- Are workflows better than declarative models in preventing the execution of wrong paths?

These three questions are conceived in terms of modeling effort, adaptation effort (effort required to change the model when it is not able to deal with an unexpected situation), and excess flexibility. The effort is measured by the time necessary for executing the corresponding task. The level of flexibility is captured through the execution of use cases that address different degrees of flexibility. Very predictable use cases do not require flexibility and should be successfully executed by both approaches. Use cases that represent more unexpected situations test for the presence of flexibility, while use cases that present wrong paths tests for the excess of flexibility.

The unexpected situations are use cases that are likely to be foreseen only by experts in the business. The experiment subjects are students with no previous experience with business process. Therefore, although these situations were known in advance by us, there was a large probability of them being unexpected by the modelers.

3.1 Experiment Design

In the context of controlled experiments design guidelines (Wohlin et al., 2000), we set up our experiment as a *single factor* (business process requirements) with *two treatments* (Workflow or Declarative modeling). The experiment was conducted in the second semester of 2011.

The experiment was designed as follows:

Subjects: the subjects of this experiment are post-graduating students of the Center for Informatics (CIn) at Federal University of Pernambuco (UFPE), Brazil. Forty students from a BPM discipline where initially selected. These students had no previous experience with business process modeling. They received training in each of the two approaches. After the training stage, we evaluate their knowledge in each approach by tests. The subjects were then split into two groups based on these tests results.

Object: a single process is implemented by the subjects. We selected a set of requirements from a software development company. This is a field recognized to be complex and highly uncertain. A group of three specialists validated the requirements. They analyzed if the requirements are fair enough to be modeled in both approaches. Additionally, five test cases were created to verify the degree of flexibility of the models. The test cases 1, 2, and 3 are obligation scenarios, i.e., represent situations which models should accept. These cases present increasingly degree of unexpectedness. The last two test cases are prohibition scenarios, i.e., represent unacceptable situations according to the business process requirements.

Metrics: We used three types of metrics to support the comparison between the approaches. According to Golden and Powell (Golden and Powell, 2000), flexibility is the property of being able to adapt. This notion can be explained by two dimensions: 1) time demanded to adapt to unforeseen situations; 2) range of options available in responding to environmental change. We assume that the second dimension impacts the first one since models that provide a wider range of possibilities need less changes, and so, they are faster to adapt. Therefore, a flexible model can be described as a model which people can modify in relatively short order (Leonardi, 2011). Based on this flexibility concept, we selected the metrics below:

- **Correctness Metrics:** indicate whether the designed model is suitable or not to the business process requirements;
- **Modeling Effort Metrics:** indicate the effort demanded to model the business process;
- **Flexibility Metrics:** indicate whether the designed model is easy to adapt in order to cope with unexpected situations (test cases);

Dependent Variables: We measure the efforts required by each approach in terms of the variables shown in Table 1. Each variable has two versions, one for the workflow group and other for the declarative group. When we need to differentiate these groups, we add a subscripted 'W' or 'D' letter to the variable symbol, referring to Workflow values or Declarative

Table 1: Dependent variables.

Symbol	Variable	Description
RCT	Ratio of Correct Requirements	Percentage of requirements that were correctly included in each model.
DT	Drafting Time	Time to study the requirements and create a draft of the process model.
MT	Modeling Time	Time to implement the model using the software tool.
TA_n	Time to Adapt the Model to Test Case n	Time to modify the model if it does not correctly process the test case *n*, where $n = 1, \ldots, 5$.

Table 2: Hypotheses.

Null Hypothesis	Alternative Hypothesis
$H1_0 : RCT_D \simeq RCT_W$	$H1_1 : RCT_D \neq RCT_W$
$H2_0 : DT_D \simeq DT_W$	$H2_1 : DT_D \neq DT_W$
$H3_0 : MT_D \simeq MT_W$	$H3_1 : MT_D \neq MT_W$
$H4_0 : TA_{1D} \simeq TA_{1W}$	$H4_1 : TA_{1D} \neq TA_{1W}$
$H5_0 : TA_{2D} \simeq TA_{2W}$	$H5_1 : TA_{2D} \neq TA_{2W}$
$H6_0 : TA_{3D} \simeq TA_{3W}$	$H6_1 : TA_{3D} \neq TA_{3W}$
$H7_0 : TA_{4D} \simeq TA_{4W}$	$H7_1 : TA_{4D} \neq TA_{4W}$
$H8_0 : TA_{5D} \simeq TA_{5W}$	$H8_1 : TA_{5D} \neq TA_{5W}$

values, respectively.

Treatment: The experiment has two treatments: a) modeling the process using the BizAgi tool, corresponding to the Workflow approach; b) modeling the process using the Kinetic tool, corresponding to the Declarative approach.

Instrumentation: We used a laboratory with the tools preinstalled to conduct the experiments. At the beginning of the experiment, the subjects received print copies of the requirements. To measure their performance, the students used a chronometer software and an electronic spreadsheet to annotate their modeling times. Along the experiment, supervisors observed the correct use of these tools by the students.

Hypothesis Definition: The hypotheses defined in Table 2 are being tested in this experiment.

3.2 Threats to Validity

Internal Validity: The participants selected for this experiment had no previous knowledge on workflow modeling or declarative modeling. They received 12h of training in each approach. By this way, we avoid any bias caused by prior experiences with an approach or another. After the training, we applied a test to evaluate the knowledge of the students. We employed statistical methods to define two classes of students according to the approach they performed better in the test. Firstly, we assigned points to each right answer in the tests, which defined their grade in the test. Then, we classified the students into four ranks, according to the quartiles of the distribution of their grades. Finally, we combined the ranks in each approach and mapped this combined rank to the interval [-1;1]. Students that were equally capable in both approaches are those who got placed near the point zero. Students that showed tendency to the declarative approach got placed over the negative interval, while students with tendency to the workflow approach got placed over the positive interval. In this process, approximately 50% of the students lied over the negative interval and other 50% lied over the positive interval, showing no predetermined tendency to any of the approaches. This enabled us to split the subjects into two highly balanced groups.

External Validity: a concern in the experiment is that the results may be applicable only to the subjects that participated in the experiment or for the tools used. We consider that the results are valid for people not experienced with process modeling. We cannot draw conclusions on whether an expert in workflow can outperform an expert in declarative modeling or vice-versa. Anyway, as the subjects have no previous experience, we can have a better grasp of the difficulties in modeling processes with one approach or the other, which could be hidden by the knowledge of experts.

Construct Validity: we designed the experiment so that we can separate the task of creating the models from the task of implementing these models in a modeling tool. This was necessary in order to separate the effects of using a particular tool for process modeling from the effects of using the modeling approach itself. Therefore, we measured *drafting* time separated from *model implementation* time. The drafts were designed by the subjects using pen and papers and were limited to the constructs available in each modeling approach. This also reduces influences from the ability of the subjects to manipulate that particular software tool. Our concern is to assure that only the choice of approaches is influencing the results, not the choice of tools.

Conclusion Validity: we employ well-known *t-test* in order to compare the significance of the differences observed between the variables measured in each treatment. By this way, we avoid drawing conclusions when the data is not statistically significant.

4 EXPERIMENT EXECUTION

We explained the scenario of the software development company used as source for the requirements. We explained the process' objectives and provided the roles involved. Each student received their own printed copy of the business requirements. We asked the students to read the document and then start to model a draft of the process in paper. The drafting time (DT) measures the time required to create such draft model. Each student that finished the draft was asked to start modeling the draft in the modeling software. The modeling time (MT) measures the time required to translate the draft model into the modeling software format.

The experiment continued by giving each student a printed description of test cases to evaluate their model. When the model did not pass on a test case, we measured the time to fix the problem.

4.1 Data Analysis

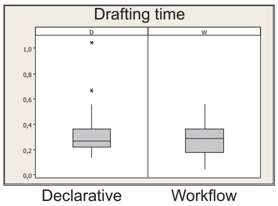
We computed the mean and standard deviation of the dependent variables. We employed the *t-test* with 95% to compare the workflow and declarative approaches and evaluate the hypotheses defined for this experiment.

5 RESULTS

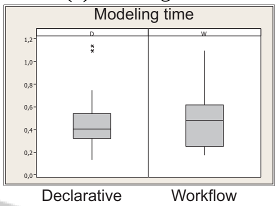
This section presents the results obtained from this experiment. We describe the statistical measures collected during the experiment and discuss the results.

5.1 Model Correctness

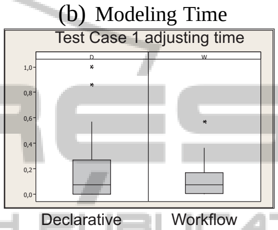
We extracted fourteen requirements from the business process description. The number of requirements the model fulfills defines its level of correctness. The list of requirements comprises one initial activity requirement (determining which activity should start the process), eight obligations and five prohibitions. To be 100% accurate, a model should fulfill all fourteen requirements. The results are as follows:



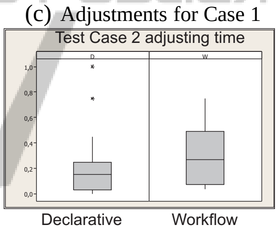
(a) Drafting Time



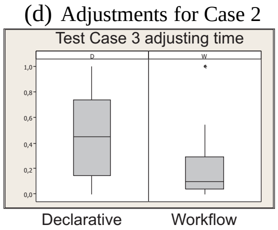
(b) Modeling Time



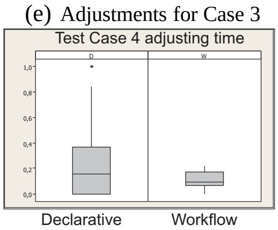
(c) Adjustments for Case 1



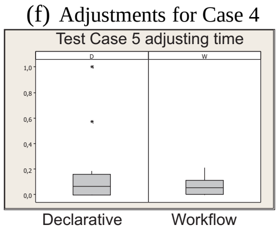
(d) Adjustments for Case 2



(e) Adjustments for Case 3



(f) Adjustments for Case 4



(g) Adjustments for Case 5

Figure 1: Comparison of measures.

- **Workflow:** the Workflow group adequately modeled a number of 9.38 requirements in average, or 67% of total requirements;
- **Declarative:** the Declarative group correctly modeled 8.4 requirements in average, or 60% of total requirements.

Notice that these numbers reflect the average quality of the models, not the number of correct models.

We compared the distributions of the total number of correct requirements in each group using the Student's t-test with 95% confidence. The test showed no statistical difference between the two approaches. Therefore, the null hypothesis $H1_0$ could not be rejected. This indicates that no advantage to a particular approach is observed with respect to the model correctness analysis.

5.2 Modeling Effort

We measured the modeling effort through the time required to write the model draft plus the time to translate it into the corresponding software representation. The results showed no statistically significant difference between the two approaches (see Fig. 1(a) and Fig. 1(b)). The average time required to write the draft for the workflow model was 12minutes (standard deviation of 6 minutes), while the declarative version had an average time of 15 minutes (standard deviation of 9 minutes). Using the tools, the modeling time for Workflows was of 54 minutes (std. dev.: 28 min.), while for Declarative models was of 58 minutes (std. dev.: 28 min.). As the results presented statistical equivalence both between the drafting times and between the modeling times, we can infer that the tools employed by this study had no influence on the comparison. Therefore, hypotheses $H2_0$ and $H3_0$ were both accepted with 95% confidence.

5.3 Flexibility

We analyzed the flexibility through the execution of test cases. These test cases were incrementally applied, i.e., the next test was applied only after the current test case had passed. Hence, it is anticipated that, while correcting a model in order to pass in a certain test case, the participant could hasten corrections of problems that would be found in upcoming test cases.

In this study, we measure flexibility through the time necessary to adapt the model to deal with unexpected situations (Leonardi, 2011). Figures 1(c) through 1(g) show the measures that reflect the effort to adapt the models to each test case imposed to the subjects. The mean and standard deviation of these variables are described in Table 3. Among these five variables, the TA_5 is the only one which presents a statistical difference between workflow and declarative approach. The efforts employed by the two groups to adapt their models to the other test cases (1, 2, 4,and 5) are not statistically distinguishable. Regarding the flexibility, hypothesis $H4_0$, $H5_0$, $H7_0$, and $H8_0$ were accepted. Only hypothesis $H6_0$ was *rejected* with 95% confidence.

These results indicate that the approach taken has low influence on the effort to achieve the process flexibility.

To further investigate the circumstances at which these results were obtained, let us discuss the tests in more detail.

We conceived the five test cases to test different levels of flexibility. Case 1 is the most predictable scenario. All models should have equal chance to successfully process this case. Cases 2 and 3 present more unpredictable scenarios. Models that successfully cope with these two or that require lower efforts to be adapted to cope with them are considered to be more flexible. Cases 4 and 5 present wrong scenarios, i.e., scenarios where the process requirements are violated. Models should not let cases 4 and 5 pass the test. Models that do let are considered excessively flexible. They also need to be adapted to correct the failures that let these wrong scenarios to occur.

Table 3: Flexibility variables measures.

Variable	Mean	Std. Dev.	Hyp.
$\mathbf{TA_{1W}}$	0:03:21	0:04:04	$H4_0$
$\mathbf{TA_{1D}}$	0:04:07	0:05:35	*not rejec.*
$\mathbf{TA_{2W}}$	0:11:39	0:09:08	$H5_0$
$\mathbf{TA_{2D}}$	0:05:50	0:05:01	*not rejec.*
$\mathbf{TA_{3W}}$	0:03:03	0:03:12	$H6_0$
$\mathbf{TA_{3D}}$	0:08:30	0:06:34	**rejected**
$\mathbf{TA_{4W}}$	0:05:02	0:03:27	$H7_0$
$\mathbf{TA_{4D}}$	0:09:32	0:11:46	*not rejec.*
$\mathbf{TA_{5W}}$	0:05:04	0:04:54	$H8_0$
$\mathbf{TA_{5D}}$	0:04:43	0:05:27	*not rejec.*

Figure 2 shows a graphical comparison of the number of models that correctly handled the situations proposed (passing on test cases 1 through 3 and not letting cases 4 and 5 pass).

Observe that test case 1 tests for a very common situation in the process. But, as this was the first test of the models, they were expected to show initial errors. Declarative models were more successful at this stage, with 30% of models being correct, against only

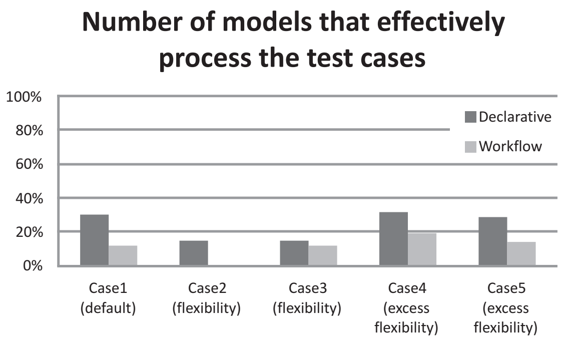


Figure 2: Flexibility tests.

11.8% of the Workflow models.

Test case 2 presents a situation with higher probability of being unexpected. This assumption showed to be true. Although the subjects had adjusted their models to case 1, most subjects failed to foresee the scenario of case 2. As a result, the success ratio dropped off significantly, with only 15% of correct Declarative models and no correct Workflow models. Similar situation occurred for test case 3. But this time, Workflows and Declarative models presented closer results.

Regarding the excess of flexibility, Figure 2 shows that the Declarative group also achieved better results than the Workflow group at test cases 4 and 5. As the Workflow group made many adjustments to make their models more flexible, they also suffered from excess of flexibility, which is a counter-intuitive result given that Workflows are presumed to be too rigid.

Despite the Declarative group showed higher quality in these tests when compared to the Workflow group, all models required adjustments. However, the efforts required to adjust the models do not reflect the differences perceived in the accuracy of the models. Indeed, the efforts were equivalent even if the Workflow models were less adequate to process the test cases than the Declarative models. In the single case where the quality of the models match (test case 3), the Workflows required significantly less effort to adjust than the Declarative models.

These results suggest that Workflow models are easier to adjust than Declarative models.

6 RELATED WORK

Several works discuss the benefits of using alternative, more flexible approaches to business process modeling (Pesic, 2008; Lu and Sadiq, 2007; Nurcan, 2008).

Reijers et al. (H. Reijers and van der Aalst, 2003) perform a comparison between the features provided by traditional Workflows and by Case-Handling Systems (CHS), which are a data-driven, case centric approach to business process design. In CHS, the employees are able to skip and redo activities and only a preferred execution flow needs to be modeled. After a thorough review of workflow and CHS characteristics, the authors conclude that CHS are not universally superior to Workflows, and that a number of extensions to current workflow technology would make it equate CHS benefits.

Few works conduct experiments for quantitatively comparing flexible approaches among each other or against Workflow. Mutschler et al. (Mutschler et al., 2008) present an experiment for comparing CHS and Workflow technologies. The results of their experiment indicate that Workflow technology requires less effort for business process implementation when compared to CHS. Moreover, subsequent process changes require significantly less effort than the initial implementation. However, no general conclusion can be derived about the strengths and weaknesses of Workflow and CHS technologies.

Weber et al. (Weber et al., 2009) conduct an experiment where they measure the capability of end-users to cope with increasing number of constraints in declarative process models. Using the Alaska simulator as the declarative process modeling and execution tool, they found no evidence of performance difference when the experiment subjects had to handle substantially varying levels of constraints. Pichler et al. (Pichler et al., 2011) compare imperative and declarative languages with respect to process model understanding. Their study finds that imperative modeling languages are easier to understand.

7 SUMMARY AND CONCLUSIONS

This paper describes a controlled experiment conducted to compare two approaches for business process modeling: workflow and declarative. The subjects had the task of modeling a business process from a set of requirements obtained from a real company. A total of 40 students participated in the experiment.

Although proponents of the declarative approach claim that declarative models are better prepared to deal with situations not expected by the modeler, no evidence to support this claim was observed in our experiment. Indeed, Workflows showed to require less effort to adapt to at least one of the test cases proposed during the experiment.

It is worth notice that related work comparing other features of flexible business process technologies and Workflows also did not find significative differences between the approaches (Weber et al., 2009; Mutschler et al., 2008) or did find an advantage in favor of the imperative approach (Pichler et al., 2011).

A strength of the present study is the high level of balance between the two groups of subjects that participated in the experiment, highly reducing any bias caused by differences in the expertise of the participants of each group. Furthermore, we removed bias from the use of the selected software tools, by asking the subjects to design the models on a paper before implementing them on the tool and measuring their efforts in both tasks.

A limitation of the experiment can be related to the size of the scenario. Once the complexity of the scenario is limited, it may be the case that the Workflow approach has not showed its most prominent problems as argued by flexibility researchers. Nevertheless, we noticed that, as the scenario becomes more complex, both approaches require more maintenance effort. Students that use the Declarative models often report difficulties in dealing with the growth of the number of rules, which is something that is likely to occur in real settings. Thus, we foresee that more complex scenarios would not benefit the Declarative approach.

ACKNOWLEDGEMENTS

This work was partly supported by the Brazilian Research Council (CNPq), grants PQ 314539/2009-3 and GD 140512/2009-8.

REFERENCES

Coalition, W. M. (1995). WfMC standards: The workflow reference model, version 1.1.

Dadam, P., Reichert, M., Rinderle, S., Jurisch, M., Acker, H., Göser, K., Kreher, U., and Lauer, M. (2008). Towards truly flexible and adaptive process-aware information systems. In Kaschek, R., Kop, C., Steinberger, C., and Fliedl, G., editors, *UNISCON*, volume 5 of *Lecture Notes in Business Information Processing*, pages 72–83. Springer.

Golden, W. and Powell, P. (2000). Towards a definition of flexibility: in search of the holy grail? *Omega*, 28:373–384.

H. Reijers, J. R. and van der Aalst, W. (2003). The case handling case. *International Journal of Cooperative Information Systems*, 12(3):365–391.

Jeston, J. and Nelis, J. (2006). *Business Process Management : Practical Guidelines to Successful Implementations*. Elsevier/Butterworth-Heinemann, Amsterdam.

Leonardi, P. M. (2011). When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies. *MIS Quarterly*, 35(1):147–167.

Lu, R. and Sadiq, S. (2007). A survey of comparative business process modeling approaches. In *In Proceedings 10th International Conference on Business Information Systems (BIS), number 4439 in LNCS*, pages 82–94. Springer.

Modeler, B. P. (2010). Bizagi: Bpmn software.

Mutschler, B., Weber, B., and Reichert, M. (2008). Workflow management versus case handling: results from a controlled software experiment. In *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pages 82–89, New York, NY, USA. ACM.

Nurcan, S. (2008). A survey on the flexibility requirements related to business processes and modeling artifacts. In *HICSS '08: Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, page 378, Washington, DC, USA. IEEE Computer Society.

Pesic, M. (2008). *Constraint-Based Workflow Management Systems: Shifting Control to Users*. PhD thesis, Technische Universiteit Eindhoven, Eindhoven, The Netherlands.

Pesic, M. and van der Aalst, W. M. P. (2006). A declarative approach for flexible business processes management. In Eder, J. and Dustdar, S., editors, *Business Process Management Workshops*, volume 4103 of *Lecture Notes in Computer Science*, pages 169–180. Springer.

Pichler, P., Weber, B., Zugal, S., Pinggera, J., Mendling, J., and Reijers, H. A. (2011). Imperative versus declarative process modeling languages: An empirical investigation. In *BPM 2011 Workshops, Part I, LNBIP 99*. Springer-Verlag.

Weber, B., Reijers, H. A., Zugal, S., and Wild, W. (2009). The Declarative Approach to Business Process Execution: An Empirical Test. In *Proc. CAiSE '09*, pages 270–285.

WFMC (2002). Workflow management coalition workflow standard: Workflow process definition interface – XML process definition language (XPDL) (WFMC-TC-1025). Technical report, Workflow Management Coalition, Lighthouse Point, Florida, USA.

White, S. A. (2006). Introduction to bpmn. Technical report, IBM Software Group.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2000). *Experimentation in software engineering: an introduction*. Kluwer Academic Publishers, Norwell, MA, USA.

zur Muehlen, M. (2002). *Workflow-Based Process Controlling : Foundation, Design, and Application of Workflow-driven Process Information Systems*. Logos Verlag, Berlin.