Comparative Evaluation of the Rule Based Approach to Representation of Adaptation Logics

Lauma Jokste
Information Technology Institute
Riga Technical University
Riga, Latvia
lauma.jokste@rtu.lv

Abstract—**Due to the rapid growth of business processes digitalization, enterprise applications cover more and more business and daily life functions thus becoming more complex. Complex enterprise applications often deal with low users' satisfaction of usability. This problem can be solved by implementing adaptation algorithms in enterprise applications, so they can be adjusted for specific context situations and specific users' needs. Some adaptation logics representation techniques are complex and require specific knowledge and skills to manage and modify adaptation process. In this paper rule based adaptation approach is introduced where rules are used as means to manage and modify adaptation process. Rules are easy to read and understand, thereby rule based adaptation should ensure elastic, transparent and easy administrable adaptation process. The goal of this paper is to test this statement by carrying out a comparative adaptation logics representation evaluation experiment. During the experiment participants are required to complete tasks which include different forms of adaptation logics representation (code, rules and models). Experiment results are analyzed by qualitative and quantitative measures such as users' understandability of applications behavior when adaptation case occurs and users' satisfaction with adaptation logics representation. Experiment results are summarized and are to be used for further development of the study.**

Keywords—Adaptation process evaluation, Adaptive enterprise applications, Adaptivity, Rule based adaptation.

I. Introduction

Adaptation can be defined as any process which modifies or extends the behavior of the system in order to improve its interactions with the surrounding parts of the system [1]. Adaptation of enterprise applications is one of the solutions how to improve applications' performance [2] and users' satisfaction [3]. Adaptation algorithms can be represented and implemented in different forms, e.g. software code, models and business rules.

McKinley et al. [4] summarize different software tools and technologies which allow to write self-modifying code and develop built-in adaptation mechanisms. For such systems adaptation code can be implemented in applications middleware level. Such approach ensures transparent adaptation process but leads to limitation that this approach can be used only if applications are written against the specific middleware platform. Adaptation algorithms can be also implemented in applications functional code, but this approach leads to the complex functional code management.

Models can help to bridge the gap between developers and stakeholders as they can be understood by stakeholders without specific IT knowledge as well as can be executed by system components typically in model driven development approaches [5]. Understandability level can be increased by using text-based syntaxes in models thought developers prefer them to graphical editors [6].

Rule based systems development began in the 1960's [7] and are still evolving in different directions such as expert systems [8], software configuration management systems [9] and adaptation systems [10]. Rules are attributed with a lot of advantages mainly related to their form which is easy to read, easy to understand by both – humans and computer system components, easy to modify and manage [11] – [13]. Rule based adaptation systems also have some shortcomings and limitations. Zacharias in [14] reports rule based system developers survey results which outline problems related to tool support for such systems development, debugging difficulties and run-time performance problems. Also problems with managing large amount of rules and dealing with conflicting rules should be considered when developing rule based adaptation systems [15]. Each adaptation logics representation format has some advantages and disadvantages. Adaptation logic expressed in a form of code is easier to develop and implement, while models and business rules improve communication between humans and software components, but often deal with implementation and development difficulties. In this paper emphasis is put on rule based adaptation by evaluating such adaptation approach's comprehension and ease of use in comparison of other adaptation logic representation

Print ISSN 1691-5402
Online ISSN 2256-070X
http://dx.doi.org/10.17770/etr2019vol2.4156

Published by Rezekne Academy of Technologies.

formats. The objective of this paper is to report the results of an experiment during which comparative evaluation of different adaptation logics representation (code, models and business rules) was carried out. Experiment results are analyzed by qualitative and quantitative measures.

The paper is structured as follows: Section II describes the rule based adaptation approach which ensures adaptation process in a form of business rules. In section III experiment design is given. Section IV summarizes experiment results and in Section V concluding remarks and future challenges are discussed.

II. Rule based adaptation approach

We have developed a context-aware rule based adaptation approach [16] which combines context monitoring, defining software entities dependencies of contextual situations data and software entities adaptation in a form of business rules. For applications adaptation purposes a separate adaptation administration module can be built and integrated with enterprise applications. This approach does not require interfering in applications code thus ensuring independent and easy administrable adaptation process.

Adaptation process includes two different types of adaptation rules: context dependency rules describe software entities dependencies of context data and can be used to determine the amount of context data that needs to be monitored and stored for further processing in adaptation process. Context dependency rules can be derived from context sources by data mining methods. Adaptation rules define an adaptation task which should be executed when particular context situation occurs. Main concepts for this adaptation approach are summarized in concepts model shown in figure 1. Adaptation rules allow to manage adaptation process by defining new rules, modifying existing rules, deleting rules or changing rules weights. Adaptation rules can be manually defined and when there will be accumulated enough historical data about adaptation rules performance, machine learning methods can be applied to generate new adaptation rules automatically.

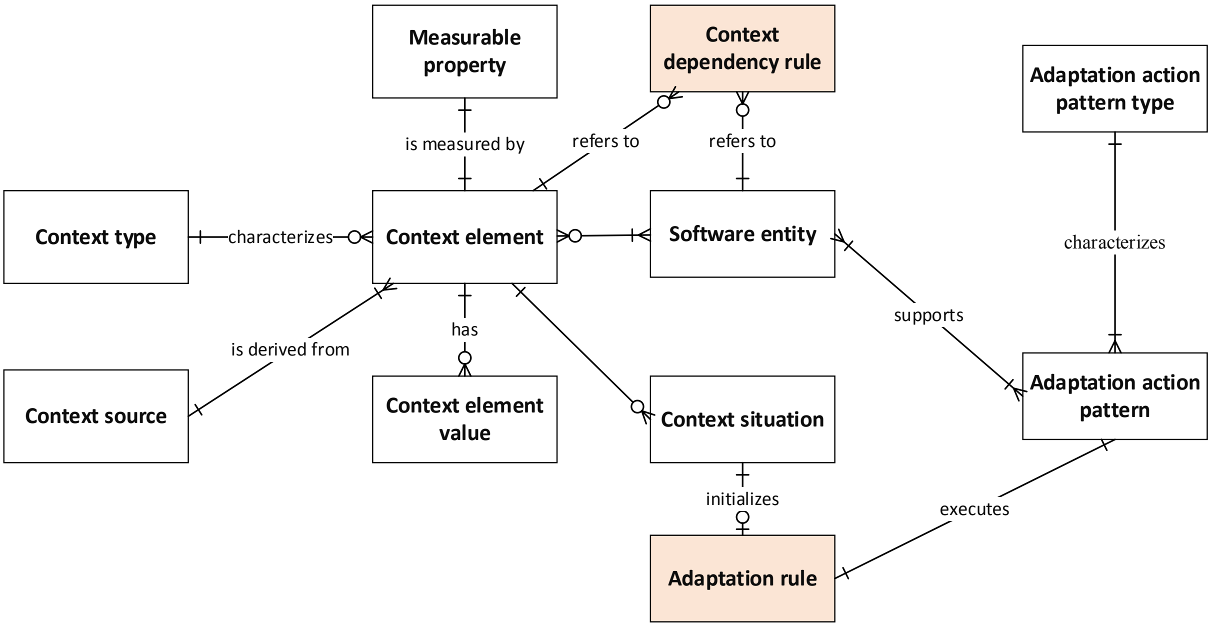


Fig. 1 Context-aware rule based adaptation approach concepts model.

In the proposed approach semantic form of event-condition-action rules is used for adaptation rules. Each adaptation rule can be described as *IF (context situation) THEN (adaptive action)*. Adaptation rules also include parameters such as adaptive action input data and expected user action. Adaptation rule concepts model is shown in figure 2.

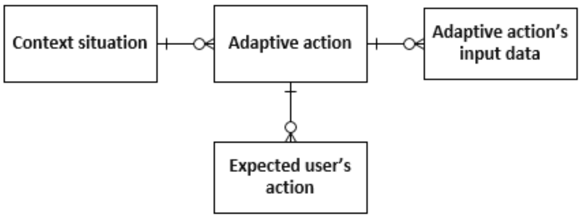


Fig. 2 Adaptation rule concepts model.

Expected user action concept allows to evaluate if adaptation rule execution gives an expected result. After executing adaptation rule, adaptation algorithm searches in data source (typically users' action logs) if user have completed expected action in expected time period. Depending on the search results, adaptation rule weight is recalculated thus ensuring automatic adaptation rules quality management.

III. Experiment design

An experiment was carried out to test the hypothesis regarding the rule based adaptation: Rules as adaptation format ensure transparent, elastic, easy understandable, modifiable and manageable adaptation process which can be administered by users without specific IT skills. The goal of the experiment was to test how different user groups understand different adaptation logic representation formats and how they evaluate simplicity and perceptibility of these formats.

The experiment was developed using online survey tool. It consisted of 6 different tasks. In each task an adaptation case was described in different adaptation logics representation form – software code, model or business rule (see task examples in figures 3 – 5). Each adaptation task included context situation in which adaptation task should be initiated and executed in application with given adaptation action input data. For each task participants were asked to: (1) describe in free text form applications behavior when adaptation case occurs; (2) evaluate how easy it was to understand applications behavior in given adaptation task representation format. For (1) participants answers were later analyzed by qualitative measures and was rated by 3 points system where 0 refers to complete incomprehensibility, 1 refers to partial comprehensibility and 2 refers to complete comprehensibility of adaptation task. For (2) a linear Likert scale was used with values from 1 to 5 where 1 refers to strongly disagree that adaptation case was easy comprehensible and 5 refers to strongly agree that adaptation case was easy comprehensible.

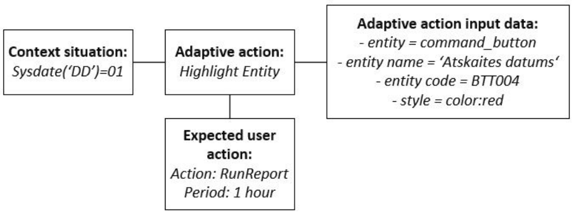
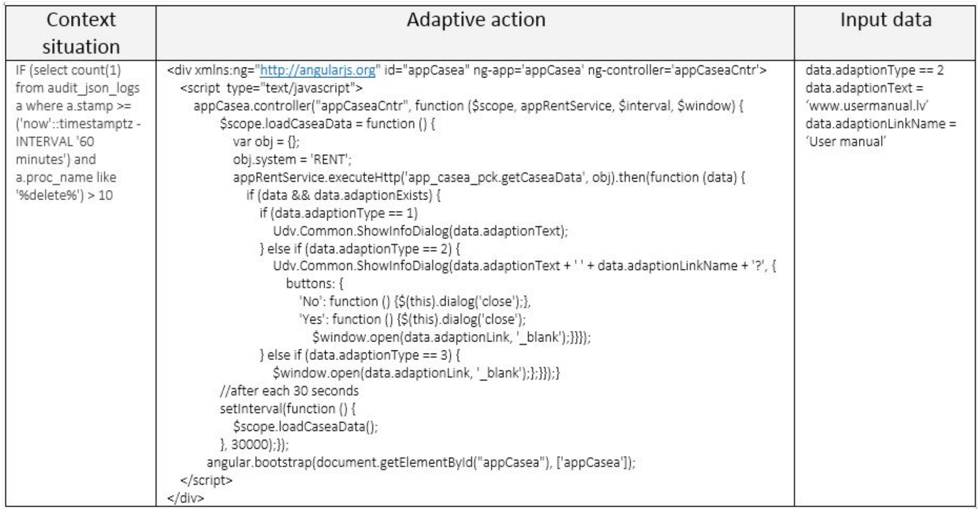


Fig. 3 Adaptation case example – adaptation rule.

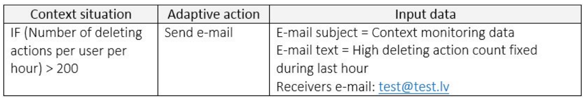
In total 20 participants completed the experiment. Testable hypothesis determines that rule based adaptation is easier to understand and manage by users without specific IT knowledge and skills because adaptation cases are described in natural language form – business rules. Accordingly 10 participants were students and/or specialists from IT field (further group_1) and 10 participants were specialists from other fields without specific IT knowledge and skills (further group_2).



Context situation	Adaptive action	Input data

IF (select count(1) from audit_json_logs a where a.stamp >= ('now'::timestamptz - INTERVAL '60 minutes') and a.proc_name like '%delete%') > 10	<div xmlns:ng="http://angularjs.org" id="appCasea" ng-app='appCasea' ng-controller='appCaseaCntr'>
	  <script type="text/javascript">
	    appCasea.controller("appCaseaCntr", function ($scope, appRentService, $interval, $window) {
	        $scope.loadCaseaData = function () {
	          var obj = {};
	          obj.system = 'RENT';
	          appRentService.executeHttp('app_casea_pck.getCaseaData', obj).then(function (data) {
	            if (data && data.adaptionExists) {
	              if (data.adaptionType == 1)
	                Udv.Common.ShowInfoDialog(data.adaptionText);
	              } else if (data.adaptionType == 2) {
	                Udv.Common.ShowInfoDialog(data.adaptionText + ' ' + data.adaptionLinkName + '?', {
	                  buttons: {
	                    'No': function () {$(this).dialog('close');},
	                    'Yes': function () {$(this).dialog('close');
	                      $window.open(data.adaptionLink, '_blank');}}});
	              } else if (data.adaptionType == 3) {
	                $window.open(data.adaptionLink, '_blank');};}});}
	        //after each 30 seconds
	        setInterval(function () {
	          $scope.loadCaseaData();
	        }, 30000);});
	      angular.bootstrap(document.getElementById("appCasea"), ['appCasea']);
	  </script>
	</div>
		data.adaptionType == 2
		data.adaptionText = 'www.usermanual.lv'
		data.adaptionLinkName = 'User manual'

Fig. 4 Adaptation case example – software code



Context situation	Adaptive action	Input data
IF (Number of deleting actions per user per hour) > 200	Send e-mail	E-mail subject = Context monitoring data E-mail text = High deleting action count fixed during last hour Receivers e-mail: test@test.lv

Fig. 5 Adaptation case example – adaptation rule.

Online experiment tool also registered the time each user spent on execution of the experiment with a purpose of determining whether there is difference between two participant groups regarding the time needed to be used to understand and complete adaptation task.

IV. EXPERIMENT RESULTS

It took on average 18 minutes to complete the experiment (group_1 average 21 minutes and group_2 average 15 minutes). Difference between groups regarding average time spent to complete the experiment can be explained by the fact that participants with IT skills and knowledge tried to understood adaptation tasks in form of software code while participants without IT knowledge mostly marked they do not understand the adaptation task and did not spent time on examining the software code.

In each task participants were asked to describe applications behavior based on given adaptation case. Each answer was subsequently analyzed by qualitative measures and rated from 0 – 2 where 0 – complete incomprehensibility, 1 – partial comprehensibility and 2 – complete comprehensibility. Mean scores for each adaptation format were calculated and results are summarized in figure 6. Results showed that group_1 demonstrated a good understanding of software code and participants were able to describe applications behavior when defined context situation will occur and adaptation task will be executed. Adaptation cases in a form of models created a lot of confusion for both groups while all participants demonstrated very good understanding of adaptation cases described in a form of business rules.

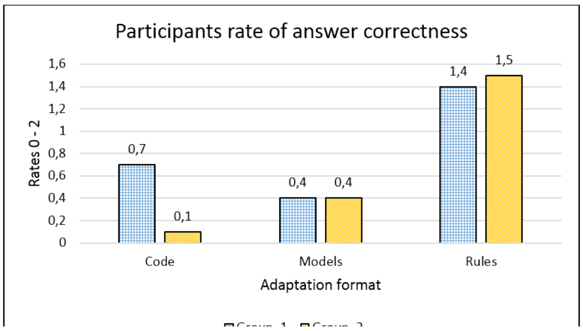


Fig. 6 Participants rate of answer correctness using linear scale 0 – 2

After completing each task of the experiment, participants were required to evaluate the comprehensibility of the given tasks adaptation logics representation format by rating their agreement to the statement *I understood applications behavior in the given adaptation case* using linear scale from 1 – 5 where 1 refers to *strongly disagree* and 5 refers to *strongly agree*. Figure 7 shows mean values of evaluation results.

Results showed that participants with IT knowledge (group_1) evaluated adaptation tasks in a form of code more understandable and simple than participants without specific IT knowledge (group_2), but still rate was low (average 1,95 out of 5). In order to read, understand and modify built-in adaptation process, adaptation administrator should be familiar with specific programming language and development technology which limits the simplicity of such adaptation logics representation format management even for users from IT field.

Participants evaluation of adaptation models comprehensibility were average 2,4 out of 5, but the actual correctness of the answers to tasks where adaptation was described in form of models were low (see figure 6). Comprehension level of adaptation models can be dependent of modelling syntax. In order to increase the level of comprehension for adaptation models as adaptation logics representation format, adaptation tasks should also include some explanation about model components and syntax.

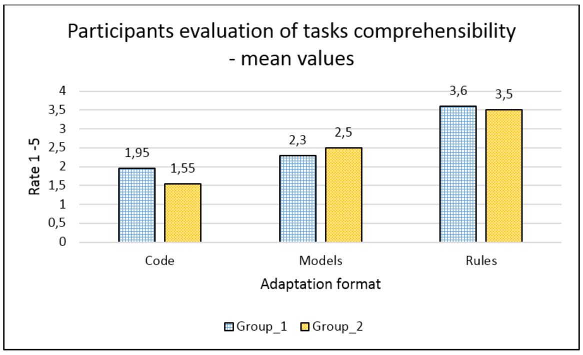


Fig. 7 Participants evaluation of tasks comprehensibility – mean values.

Experiment results showed that participants did well with tasks where adaptation logics were represented in a form of business rules. Participants demonstrated good comprehension of applications behavior when defined context situation will occur as well as highly appreciated tasks simplicity and intelligibility. Results quite slightly differed between both participant groups as business rules are described in a form close to natural language and there are no specific knowledge required to read, understand and manage them.

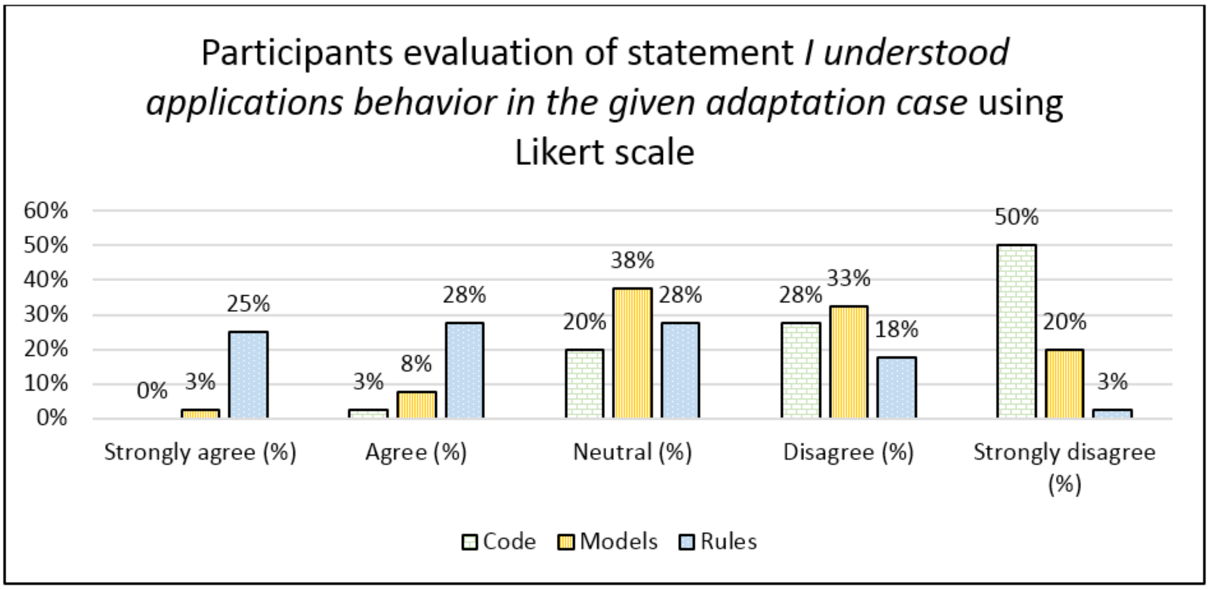


Fig. 8 Participants evaluation of statement *I understood applications behavior in the given adaptation case* using Likert scale.

In figure 8 participants agreement to the statement *I understood applications behavior in the given adaptation case* using Likert scale is shown. Diagram shows that 25% of participants strongly agreed to the statement regarding business rules while none of the participants strongly agreed to the statement regarding software code which certifies rules comprehensibility.

Overall experiment results allowed to test the hypothesis regarding rule based adaptation process transparency and comprehensibility which are important characteristics of adaptation process management. Experiment results also allowed to compare adaptation process comprehensibility for different adaptation logics representation formats.

V. CONCLUSIONS AND FUTURE WORK

In this paper we report the results of experiment during which the comprehensibility and simplicity of different adaptation logics representation formats were tested and analyzed. Results outlined an important advantage of using business rules as adaptation means for adaptive enterprise applications. Comparing to software code and adaptation models, rules demonstrate an ability for fast and simple adaptation management process since rules are expressed in natural language form thus are easy understandable for humans without any specific IT knowledge in the same time being readable and executable by software components. Nevertheless code would be more expressive than rules if complex adaptation cases should be implemented.

Despite the good characteristics of rules as adaptation means, they still lead to various difficulties such as relatively complicated development of rules execution mechanisms as well as challenges with rules run-time performance measures and assessment of the efficiency of the adaptation process.

So far in our research we have developed context aware rule based adaptation approach and tested the comprehensibility of such adaptation process. Further work will focus on rules run time performance evaluation and adaptation approach's overall evaluation regarding its effectiveness and usefulness.

References

[1] S. Kell, "A Survey of Practical Software Adaptation Techniques", Journal of Universal Computer Science, vol 14, issue 13, 2008. DOI 10.3217/jucs-014-13-2110

[2] F. D. Macias-Escriva, R. Haber, R. Toro and V. Hernandez, "Self-adaptive systems: A survey of current approaches, research challenges and applications", Expert Systems with Applications, 40(18), p. 7267-7279, 2013

[3] D. Te'eni, R. Feldman. "Performance and satisfaction in adaptive websites: an experiment on searches within a task-adapted website", Journal of the Association of Information Systems, vol. 2, Article 3, 2001.

[4] P. K. McKinley, S. M. Sadjadi, E. P. Kasten and B. K.C. Cheng, "Composing Adaptive Software", Computer, vol. 37, Issue 7, 2004, DOI: 10.1109/MC.2004.48.

[5] L. Jokste, "Integration of Business Rules and Model Driven Development", Short Paper Proceedings of the 6th IFIP WG 8.1 Working Conference on the Practice of Enterprise Modeling (PoEM 2013), vol. 1023, 2013.

[6] A. Uhl, "Model-Driven Development in the Enterprise", IEEE Software, 2008

[7] D. A. Partridge, K. M. Hussain, K. M. Hussain, "Knowledge Based Information Systems". Mc-Graw Hill book comp., 1994

[8] A. Abraham, "Rule-based Expert Systems", Handbook of Measuring System Design, edited by Peter H. Sydenham and Richard Thorn, 2005.

[9] G. Ge, E. J. Whitehead, Jr. (Advisor), "Automatic Generation of Rule-based Software Configuration Management Systems", Proceedings of 27th International Conference on Software Engineering, 2005. L. Jokste and J. Grabis "Rule Based Adaptation: Literature Review", Environment. Technology. Resources: Proceedings of the 11th International Scientific and Practical Conference, vol.2, 2017

[10] T. Zhao, H. Zhao, W. Zhang, and Z. Jin, "User Preference Based Autonomic Generation of Self-Adaptive Rules", Internet ware 2014 Proceedings of the 6th Asia-Pacific Symposium on Internet ware, p. 25-34, 2014.

[11] T. Tran, P. Cimiano and A. Ankolekar, "Rules for an Ontology based Approach to Adaptation", Proceedings of the 1st International Workshop on Semantic Media Adaptation and Personalization, 2006.

[12] J. Rubart, "Semantic Adaptation of Business Information Systems using Human-Centered Business Rule Engines", Proceedings of 10th International Conference on Semantic Computing, p. 187-193, 2016

[13] V. Zacharias, "Development and Verification of Rule Based Systems – a Survey of Developers", Lecture notes of international workshop of Rules and Rule Markup Languages of Semantic Web, 2008. https://doi.org/10.1007/978-3-540-88808-6_4.

[14] M. Zeeshan and S. A. Khan, "A novel algorithm for link adaptation using fuzzy rule based system for wideband networking waveform of SDR", International Journal of Electronics and Communications (AEÜ), vol. 69, p. 1366- 1373, 2015.

[15] L. Jokste and J. Grabis, "Context-Aware Adaption of Software Entities using Rules", ICEIS 2017: Proceedings of the 19th International Conference on Enterprise Information Systems, vol. 3, SciTePress, 2017.