Evaluation of Domain-specific Rule Generation Framework based on Usability Criteria

Conference Paper · June 2018

4 authors:

Neel Mani
Amity University

Shastri Lakshman Nimmagadda
Curtin University

Markus Helfert
Dublin City University

Torsten Reiners
Curtin University

Evaluation of Domain-specific Rule Generation Framework based on Usability Criteria

Completed Research Paper

Neel Mani
School of Computing
Dublin City University
Dublin-9, Ireland
neelmanidas@gmail.com

Shastri L Nimmagadda
School of Management
Curtin University
Bentley, Perth, WA, Australia
shastri.nimmagadda@curtin.edu.au

Markus Helfert
ADAPT Center for DCT
School of Computing
Dublin City University, Dublin-9, Ireland
markus.helfert@dcu.ie

Torsten Reiners
School of Management
Curtin University
Bentley, Perth, WA, Australia
T.Reiners@cbs.curtin.edu.au

Abstract

The challenges faced by domain experts, commitments made to domain-specific rule (DSR) languages and process design are described. We investigate the business application developments and existing challenges of evaluation strategies of DSR articulations. Often, multiple domain scenarios pose end-user predicaments complicating the computational ability of DSR. In addition, implementation of DSR and its configuration are belated due to poorly evaluated usability criteria. A new framework is needed, facilitating the DSR language and enhancing the computational intelligence. We intend to evaluate the performance of DSR generation and framework integration with variety of usability conditions including efficiency and effectiveness of configuration through system usability score (SUS). Empirical research involving experimental data, questionnaire surveys, and interview outcomes provide conclusive evaluation attributes and their fact instances from SUS. Both manual and semi-automatic configurations are tested. Semi-automatic configuration appears to be more efficient and satisfactory with regard to artefact performance, quality, learnability, user-friendly and reliability.

Keywords: User Experience, Domain-specific Rule, Usability, Variability Model, Evaluation Process and Planning

Introduction

In Computer and Information Sciences perspective, a domain is an entity or dimension controlled by a set of rules and their languages. The domain-specific rule language (DSRL) by industry standards, constitutes a set of rules to use them in a single domain, logically making process-free data, without limiting the data by any structural constraints. The usability itself is an entity or attribute dimension, and the level is narrated at which, a product or service can be used by stated domain experts to achieve particular goals with efficiency and satisfaction in a specified context". A more widely accepted definition of usability is given in (ISO 9241-11) within a purview of business outlook. Now it is updated as ISO 9241-11:2018, providing a framework in an agreeable concept of usability and attaching it to situations where human-computer interaction (HCI) and other types of business systems are dealt with in the form of either product and or service delivery. However, the business applications face huge development challenges due to rapid and diverse business contexts including dynamic and competitive market environments (Papulova and Papulova 2006). The changes at times are tenacious, and process models may need continual customization to meet new demands of end-users either in a specific domain or unpredictable multiple domains. Incorporating changes in business applications is a time-consuming task, because of the complexity in system configuration, heterogeneity and multidimensionality of data sources, availability of hundreds of software libraries and rigidity of the process models (hard-coded components). Consequently, the entire application passes through several stages, such as development, deployment of the testing environment: testing, test reporting, redeveloping including successive code testing. In many cases, the necessary changes are incorporated rapidly and implemented without pursuing any other process, which may increase the time to deliver the product/service to the market. In a standard setup, the changes should first be understood by a domain expert, who should then be able to explain the algorithms to a program developer. Then the developer elaborates the application according to the interpretation and research objective. The development process takes time, adding human resources, with beginning-to-end and end-to-end loop reparative processes. More iterations in programming can create added error prone codes; compile the code and install it on the application server are other challenges. With every code change, we need to go through the similar process and sometimes restart the server.

In contrast, the proposed solution allows a non-technical user to initiate the modifications without knowing the procedural details such as the computing algorithms and code. The design of a domain-specific system, or an application, that aims to resolve domain -specific related issues may impact the functional and operational quality (FOQ) of the final framework solution. We discuss several challenges that impact the FOQ: the first part of the work relates to the appraisal of suggested DSR configuration (Mani et al. 2017) regarding its efficiency and effectiveness. The second part focuses on the system usability score (SUS) satisfaction of the framework as judged by the end-user experience. For user experience evaluation, we standardize usability as prescribed in (ISO 9241-11:2018). Broadly, the usability in the current context refers to "which extent a product can be adopted by domain experts to achieve specific goals with framework effectiveness, efficiency, and satisfaction in a specified context of implementation". Usability is regarded as an apparent prerequisite for all types of technology (Dix 2009) affirming the usability properties with which the end-users achieve the strategic goals. The usability objectives are measured autonomously, irrespective of any particular domain of human activity, design and development of various software systems, including digital content technology (Tselios et al. 2008). The key foci for assessing the configuration of the rule generation in a framework are concomitant with usability properties. They facilitate us analyzing how beneficial the offered solution is, and how efficiently it can resolve the problems posed by end-users in real-world scenarios. The objective here is to make the solution more appropriate, in response to the real-world problems put by end-users. The efficiency of the semi-automatic (proposed) versus manual (traditional or baseline) configuration is measured by differences in their response times, as obtained by both approaches during execution. The effectiveness of rule configuration is measured in terms of error prevention and the correction needed in the planned solution. The agreement is completely subjective, and the SUS can facilitate to find alternative answers. Our approach is validated through experimental results with appropriate statistical inferences and significance.

The paper is organized into various sections. We have described the evaluation process while designing the research objectives in the current context. We have demonstrated a case study for DCT with a process of data extraction. An experimental setup is designed. Evaluation of overall framework by system usability score (SUS) is done. The conclusions and future scope of work are included.

Literature Review

Domain-specific rule language (DSRL) controls set of rules to make data more process-free in a specified domain, and it is described at length in Mani et al. (2016). The current research addresses evaluation of DSR using various qualitative and quantitative approaches. Barisic (2017) provides a framework in support of usability evaluation for DSR languages. The author deals with issues of its omissions in the development of DSRLs, the significance of practical features in DSRL articulations and framework development scenarios. The author has used USE-ME, a conceptual framework that supports the DSRL conception and implementation. Barisic et al. (2010, 2011, 2012) provide systematic software language engineering aspects with a special focus on usability evaluation of DSRL. It is an experimental validation technique, bearing a user interface to assess the impact of new rule languages, adaptable in the framework formulations. This approach supports empirical studies and controlled experiments with end-user experiences. Bellamy et al. (2010) provide programming aspects of the evaluation and usability criteria using *PLATEAU* framework. We use a conceptualized research framework in the evaluation of DSR. In the current research, we have considered an experimental setup to validate the evaluation and process planning framework through users' experience. In addition, we present various issues and challenges faced while implementing the usability criteria.

Issues and Problem Statement

Badly designed DSRL at times can lower the productivity and hard to adapt to users and or non-technical users' domain. To reduce the unintended complexity of the software development process, DSRLs are used as benchmarks, facilitating their evaluations. One of the key features of the usability is to ease the risk of reducing the productivity and improve DSRL users' performance.

Research Objectives and Process

We intend to develop the framework and evaluate it with usability experiments to meet the end-users requirements, satisfying the non-technical users' queries. Various constraints and criteria parameters that can address the domain-specific challenges are assessed. Minimizing the developmental efforts of programming syntax, compilation and customization by non-technical users is evaluated. In addition, the research are aimed at evaluating the effectiveness of the configured rules by non-technical users. We pursue the following research process merits:

1. Rigor on and develop a rule-based framework to customize and evaluate the process models.
2. Evaluate through user-experience surveys and their profiles.
3. The usability of rule-based languages by non-technical domain experts.
4. Flexibility in implementation in various domain applications. Whether the prototype or integrated framework can meet the end-users' usability requirements are evaluated.

Evaluable rules that intended for the enterprise software applications may drastically reduce the development, testing and debugging times. They are assessed by the usability criteria. We also evaluate the conceptual models used in the business process management, especially in the contexts of DSRLs and within the purview of process constraints' management.

Challenges, Motivation and Significance

One of the challenges that relate to the knowledge transfer is taking domain concept to conceptual model. The other challenge relates to the configuration of DSR in process model language. A domain-specific approach provides a dedicated solution for a defined set of problems. This research has the following challenges and contributions:

1. The first contribution is a domain-specific rule generation (DSRG) framework for translating a set of rules from high-level models (domain models) on an ad-hoc basis by the end user.
2. The framework structures the components of the feature model to provide the end user to select their requirement and customize the domain template. The customized domain template manages creating and configuring the DSR to ensure the configuration of rule by the end users is efferent, effective and has a satisfactory evaluation.
3. The manual rule created and configured is an error prone and time-consuming. The generation of rule and configuration for customization of process model is based on end-user requirements (stakeholder or domain user or customer) and their evaluations.

Evaluation Process and Planning Methodology

The goal of the research is to generate rule-based – verification of the feature and evaluate with its configurable parameters. In this context, first, we ascertain that the framework is valid and representative to real-world changes and challenges. Secondly, we evaluate the usefulness of the framework and rule configuration to end-users and how an overall framework can easily be adaptable to domain experts.

The steps for evaluation process are as follows:

- Define the evaluation strategy and criteria.
- Use of empirical case studies for evaluating the rule configuration in a particular domain.
- Conduct the user experience evaluation.

Collect data from different modes[1] of tasks (experiments) and assign tasks as a participant activity, such as the configuration time. In spite of domain constraints and parameters affecting the DSR prototype, the generated rule language can weigh the end-user configuration, experience, bringing the non-technical users within the scope of DSR framework (Mani et al. 2017). We categorize efficiency, effectiveness, interoperability and satisfaction as usability criteria and performance, processing and configuration time as sub-criteria.

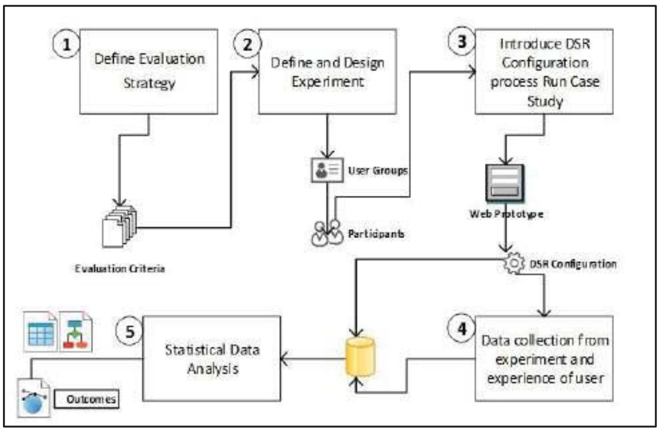

Figure 1. Evaluation Process and Planning

[1] Mode of tasks like semiautomatic, manual and SUS

The stages are detailed here:

The first step (Figure 1) of the evaluation procedure is to define criteria precisely and regulate it with an appropriate artefact. The evaluation criteria are derived in such way the generated rule is configured even without the knowledge of technical details and research output. The configurable domain-specific rule customizes the process model.

In the second step, the rule configuration change process is evaluated deploying empirical case studies in the chosen domain. We discuss the experimental setup (in terms of domain selection and rule configuration development), empirical results and analysis of the empirical results in the forthcoming sections. For practical results, statistical investigations are performed on rule configuration, shaping up an integrated framework and judging empirical case studies with a couple of dissimilar groups in Digital Content domains. Web-based user experience is carried out with usability evaluation between manual and semi-automatic configurations. We strategize tasks dividing configurations into two dissimilar categories. The dominant category is associated with the configuration of the created rule that divides the manual and semi-automatic types. Manual model is defined as a simple text editor where the participant configures the rule. Semi-automatic model is defined as text box and corresponding to each parameter, the rule needs to be configured. Each participant had two manual and two semi-automatic tasks, which were allocated automatically at the time of registration. The second category of the task is SUS. It is entirely a subjective task based on five positive and negative sentiment questionnaires. Each user has to independently finish 5 different tasks which are pre-assigned on their dashboard in the web interface after login. The need of user experience is required in the data collection for evaluating the performance of rule configuration in experiments under a controlled environment.

The third step of the valuation process of the research is to focus on the user experience evaluation. Typically, the prototype evaluation takes 20-30 minutes during registration; participants are asked about domain knowledge[2], skills, and technical knowledge. The evaluations are compared to manual and semi-automatic configurations. Additionally, it allows determining which system is better concerning the usability.

At the end of the evaluation, we collect the data for tasks, such as participant activity, configuration time, what was configured, how much time was taken for configuring the tasks. The number of errors observed while performing and configuring the tasks and the feedbacks obtained for parameters mentioned in the last stage of the evaluation process are all reviewed. This phase may be interpreted as a collection of participant’s data. Further, a combination of qualitative and quantitative methodologies (with more focus on qualitative approach) was chosen to evaluate the methodology. The SUS was adopted at this stage to collect the quantitative data as it provides a mechanism for measuring the usability satisfaction for end-users (Sauro 2011). Analysis of raw data and translating them into practical research outcome was the last phase of the evaluation process. The analysis is aimed at retrieving some relevant data which can facilitate in gauzing the issue or by examining the situational perspectives and behavior of individuals within the contextual domain (Kaplan and Duchon 1988).

Table 1. Summary of Problems, Proposed Solutions, and Evaluation Methods

Challenge	Evaluation sub-criteria	Evaluating factors	Evaluable processing criteria
C1	Efficiency	Performance	Processing time DSR Configuration
		Accuracy	Error detection
C2	Effectiveness	Quality	Error prevention

[2] Domain knowledge about Digital Content Technology – Domain knowledge is part of the design. It means, the selected participants have certain domain knowledge (Specifically, how to extract the data from different source)

C3	Satisfaction	Effectiveness	SUS
		Efficiency	
		Learnability	

Evaluation Strategy

Both qualitative and quantitative approaches are used throughout the evaluation. The evaluation scheme broadly covers the following methods:

- Case study
- Controlled user study experiments
- end-user opinions and feedback analysis (SUS)

Evaluation Criteria

The usability is evaluated by effective, efficient and satisfactory properties. The solution should conform the evaluation properties when end-users configure the domain constraints to specific conditions. As illustrated in Figure 2, we evaluate the DSR (Mani et al. 2016) and its usability criteria. One module of the ISO standard 9241 is about narrating the usability specification that applies equally to both hardware and software designs. The evaluation properties are described in the following sections.

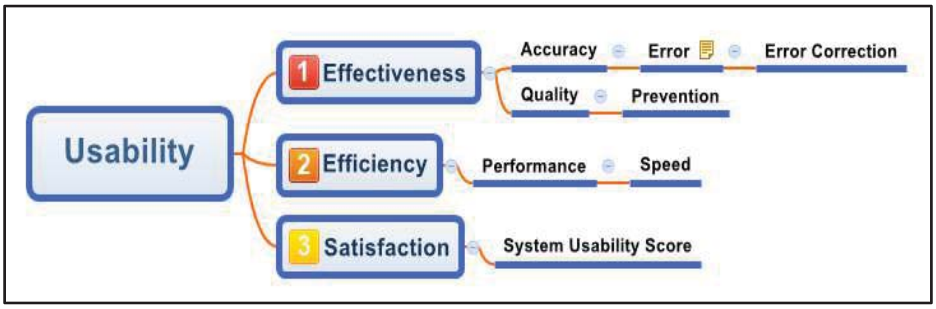

Figure 2. Usability Criteria

- Accuracy of configuration: the solution ensures error-free rule configuration and its deployment on the server. An error-free configuration helps to run the process model application smoothly, i.e., without interruption while producing accurate output. We evaluate the accuracy by analyzing the system's capability with error prevention, error correction and error message.

- Quality of configuration: quality of configuration refers to the system's capability to prevent functional, operational and data errors, such as type-, semantics-, syntactic- mismatches. In our experiments, we consider the data type of the input value as a quality parameter, in which process how many errors were prevented through dynamic validation at the time of semi-automatic configuration.

- Efficiency: It refers to ensuring that the attributes of the generated rule require minimum configuration and processing time. The processing time is estimated based on an evaluation of configuration of the constraints, and feature parameters. Later, using randomized tasks for generating rules and parametric values of different sizes, we determine the time needed to configure the rules. The configuration time is judged with the time while assigning values to the parameters by individual participants.

- Performance - The performance is measured based on configuration time that includes run-time semi-automatic and manual configuration of the rule, domain constraints and their validations. In other words, it refers to the capability of the solution providing the required performance (in terms of time), relative to the number of resources used under stated conditions. By connecting the time for rule configuration between manual and semi-automatic modes, the time taken to improve the semi-automatic configuration is measured over the traditional or manual one.

- Satisfaction: it is the support provided by the tool, allowing end-users to select features and generate rules. This includes the implementation of the rule configuration. SUS is used for an end-user intervention to evaluate the satisfaction.

The user experience as an experimental setup is felt in data collection for evaluating the performance of rule configuration in controlled environment experiments. The objective of organising porotype evaluation is compared to the manual and semi-automatic configuration. The second purpose is to obtain which system is best in terms of usability. With regard to the feature selection and tasks, the total number of feature combination is 4! (4x3x2=24). Every task is divided into two categories like manual and semi-automatic, the total number of tasks is 24x2=48. We assigned five tasks to every user and divided into two different categories. The first category of the task is to configure the generated rule; it is also divided into manual and semi-automatic tasks, the four tasks for configuration and 1 subject task SUS. We take 24 different combinations of the feature and divided into two different groups, each group has 12 users. Each user has to finish five different tasks which are pre-assigned on their dashboard in web prototype, after login authentication.

Case Study – Process for Data Extraction and Digital Content

The case study considers a scenario of customizing Digital Content Technology (DCT) service for machine translation. The DCT domain has many activities. The key process pursuits are: data extraction, segmentation and named entity recognition, machine translation, quality estimation, and post-editing. For demonstration purposes, we focus on the extraction sub-process, which is a part of the DCT business process that separates data from dissimilar sources like from text, web, document and multimedia bases (Figure 3). The data extraction is an initial and fundamental operation for retrieving data for machine translation. This process validates the research and proves which mode of configuration is better for overall framework including the usability evaluation.

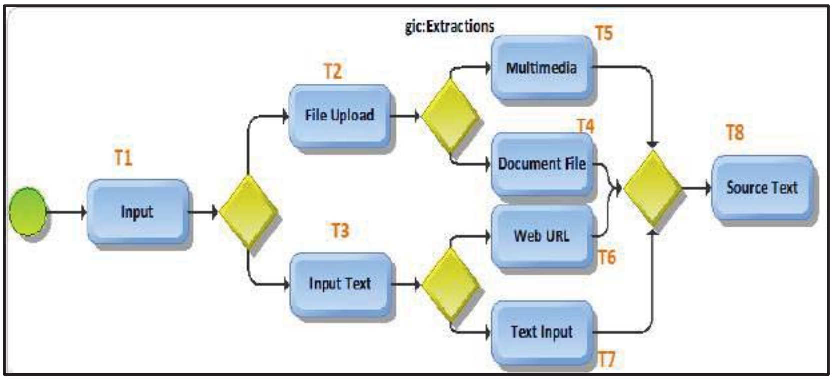

Figure 3. Extraction of Sub-Process Model in the Digital Content Technology

Further, we have made a comparative analysis of the manual and semi-automatic modes of configurations during data extraction activity. After literature review and interviews with BPM industries, a manual configuration is considered as a baseline (or traditional) system to compare the configuration with suggested semi-automatic approach. The emphasis is put on analyzing the relative benefit of the framework in the manual approach, conforming the efficiency, effectiveness, and satisfaction properties so as to achieve the operational compliance support. The feature selection and configuration scenarios involve modifications of resultant improved complex process activities that affect the function and operation of the process models.

Experimental Design

An experimental setup is made remotely as a user experiment through a web portal[3] using rule designs. The user experiment is chosen remotely to reach broader audience of domain and non-domain users within context of DCT. The benefit is that it is a controlled experiment, with fixed tasks having different modes of settings (manual and semi-automatic). There is no control over the configuration value in manual setting. Both analytical and experimental evaluation are used to assess the manual and semi-automatic configurations for performance, concerning efficiency and effectiveness. The analytical approach evaluates the performance in speed/time, accuracy in error, correctness, and user satisfaction. The prototype is thus implemented through experimental evaluation of manual and semi-automatic modes at process run-time for performance and correctness. The experiment was completed on an extraction sub-process of DCT in the real business process model situation. As suggested in Figure 3, there are 8 classes and 8 activities (T1-T8) respectively in the case study, illustrating 27 class attributes in the whole experiment.

Definition and Planning

The experimental evaluation strategizes products and teams (Basili 1996), where a researcher perceives the quality and knowledge of a product (software/technique), using a derivable set of variables for anticipated observations. We consider a number of experiments in the DCT cases to investigate and evaluate the effectiveness, and efficiency of the framework with different domain constraints and values. Table 2 provides a brief outline of the experiment.

Table 2. User Experience Evaluation Methods

Evaluation Factor	Evaluation context
Lab test	Prototyping- Framework
Field tests	Competitive evaluation of prototypes in the manual and semi-automatic environment
Field observation	Experiment result statistical analysis and observation
Evaluation of group	Evaluating result statistical user experience
Instrumented product	TRUE Tracking Real-time User Experience
Domain	Digital Content Technology
Approach	Evaluating UX jointly with usability
Evaluation data	Focus groups (multiple groups or measures, participants) evaluation(Quasi-Experiment)
User questionnaire	System Usability Scale
Human responses	PURE - preverbal user reaction evaluation
Expert evaluation	Expert evaluation Perspective-Based Inspection

The processes associated with semi-automatic performance measurement and quality assurances are vital. Different users though adopt different rules, but the manual rule configuration is chosen to statistically analyze the efficiency, concerning configuration time, and the effectiveness in terms of

[3] http://dsrl.nlplabs.org/

error propensity, accuracy, and correctness. We depend on a semi-structured questionnaire survey for collecting qualitative data. The survey comprises of several open-ended questions, pertained to usability of the system. We have added the scope and details of participants with group and user selection as described in the following sections.

Participants

In our research, more than one group measures different parameters, but the participants were not randomly assigned to different tasks. This type of experiment called quasi-experiment (Lazar et al. 2010). The study structure is factorial design because there are two or more independent variables in our research. The factorial design determines the number of conditions, so we consider adapting between-groups or within-group or split-plot. The between-groups participants are only exposed to one experiment and within groups, participants to be exposed multiple experimental conditions. We used within-group participants for multiple tasks and conditions. We select participants from ADAPT centre and other digital content institute and universities, where digital content is the main research area. Participants are practising web mining, machine translation, information retrieval areas. Additionally, the participant must have known about the digital content. A total of 20 participants completing the experiment, we divide into two different groups and compare the result of individual performance as well as group performance including subjective SUS and feedback of each participant.

Evaluation of Overall Framework by System Usability Score (SUS)

The interpretations and responses received by the participants are in accordance with the criteria set in the process of evaluation, considered as the final step in the SUS list. Figure 1 describes the collection of data based on the comments of the participants. Both qualitative and quantitative analysis techniques are deployed for validating the methodological framework. As an evaluation strategy, the logic and motives of the choice of the approach are detailed in the following sections. The respondents articulate their opinions in the last section with comments on a 5-point scale (by selecting a radio button on SUS scale), along with any specific observation made in a text box (as presented in a User interface of SUS form[4]).

System Usability Score Process

The evaluation commences with the identification and categorization of parameters or criteria. The data collection starts from the participants as per the steps described in Figure 1. The Post-Study System Usability Questionnaire (PSSUQ) (Lewis 1995) survey is an accepted tool, from which the quantitative data are acquired. The SUS serves as an evaluable instrument, since it allows capturing the instances of usability factor (Bangor et al. 2008). A total of 10 questions is presented on SUS scale, where participants have option of choosing five response categories ranging from “strongly disagree” to “agree strongly”. In addition, to understand the logic of the respondents’ response-selection, the data are gathered and analysed qualitatively from responses. It is an initial phase of data collection interpreted as “raw data”. Further, SUS answer sheets provide quantitative results and the open-ended text comments for each of SUS questions that are qualitative in nature.

SUS Calculation and Measurement

Prerequisites are acknowledging participants’ initial instant responses after evaluating each specific question, instead of taking more time about each item. Another condition is each question is answered. In case, a participant is undecided concerning a specific question, the center point of the scale is checked for that item. After the scale is properly filled, the score is obtained for the entire scale, providing a

[4] Web interface of SUS http://dsrl.nlplabs.org/UsabilityScale.aspx

composite measure of the system and its usability as a whole. Separate item scores may be unrelated in SUS. For obtaining the final score in SUS, the discrete item scores are initially determined, which range from 0 to 4. For items 1, 3, 5, 7, and 9, the score is input at a scale position "minus" 1. For items 2, 4, 6, 8 and 10, the input is 5 "minus" the scale position. In order to calculate the total score for the scale, the sum of all scores is multiplied by 2.5. The overall score should be between 0 and 100. An example of SUS scoring description is presented in forthcoming sections.

The tools under evaluation may differ, in particular the effectiveness based on search features of the web-environment and goals of evaluation (Molich and Nielsen 1990). Popular evaluation methods are heuristic (Nielsen 1994), field studies and observations (Tognazzini 1992 and Preece and Rombach 1994), filling questionnaires based on usability of the prototype and participant accomplishment in the web-based environment. Besides implicating evaluation of rule configuration systems, recognizing a design science articulation (beyond the scope of current study) is an added motivation. It involves adequate tools to evaluate the usability of different framework components in the digital content domain, besides pursuing the satisfaction and effectiveness properties of the prototype of the framework to its usability. For this purpose, an accomplished research of a digital content domain framework is needed in a controlled environment by a group of domain experts. The framework prototype developed is operational, allowing a platform, in which customization of the process model and configuration of its operational part are accomplished by non-technical domain users amid bringing a rule language in the customized domain model. The usability is evaluated accepting the SUS component of the prototype.

Experimental Results

Analysis of statistical attainment and effectiveness is provided where processing time and configuration efficacy are evaluated on the basis of the precision and value of the configured rule. The SUS is adopted for validating the statistical results and their evaluation. In order to assess the performance of the framework, the statistical score is compared with the subjective score. The subjective score of the model is described in the forthcoming section.

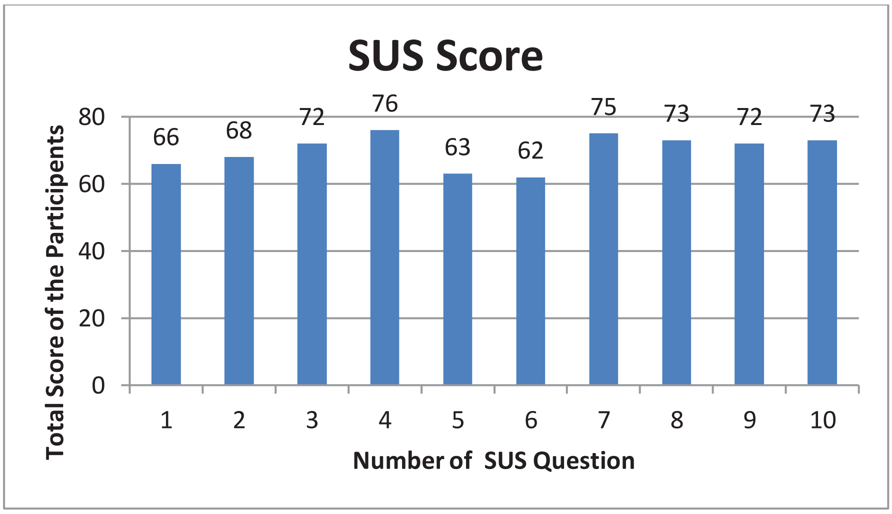

Figure 4. SUS Score Individual Questions

In addition to the quantitative analysis outlined above, a pre-defined questionnaire approach was employed with the SUS to satisfy the usability evaluation criteria discussed in Tables 1 and 3, and Figure 2. Reflecting the research criteria, three main areas of evaluation are: efficiency, effectiveness, and satisfaction. The sub criteria of efficiency is performance, i.e., processing or configuration time. It appears the system is very cumbersome to use, because it is associated with efficiency and its SUS score is 73, suggesting the prototype is efficient.

Table 3. Examples of Usability Metrics (ISO 9241: 2018)

Usability Objective	Effectiveness Measure	Efficiency Measure	Satisfaction Measure
Suitability	% achievement	Task completion time	Rating scale
User training	Error prevention	Relative efficiency	Rating scale
Error tolerance	Error prevention (%)	Debugging time	Rating scale

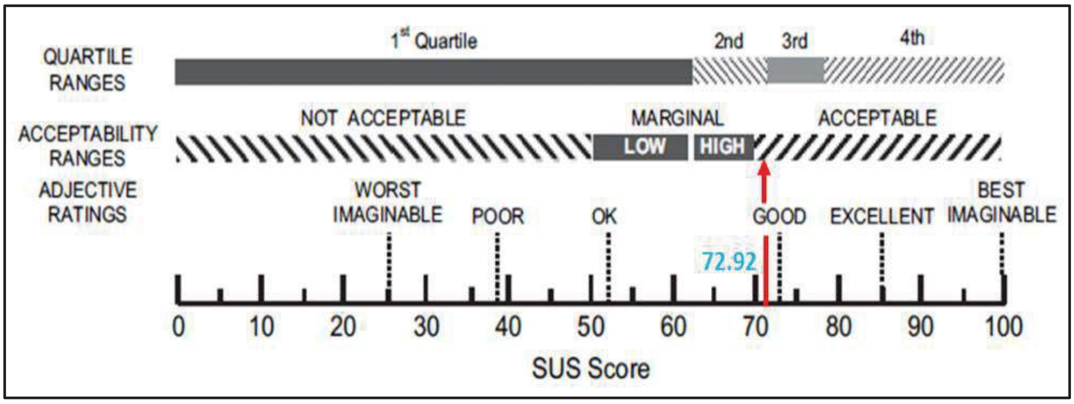

Figure 5. SUS Normal Scale

As inferred in Figures 4 and 5, it is evident that the semi-automatic configuration is more efficient, effective, and satisfactory than the manual configuration in terms of performance, accuracy, quality, learnability, user-friendliness, and reliability. The horizontal axis represents 5 positive and 5 negative questions, and as discussed in Figure 4, the vertical axis specifies the total number of points corresponding to each question.

Discussions

The effectiveness of the usability criteria and their evaluation approach depends on to a great extent on specific characteristics of the evaluated environment and the objectives of evaluation under study. The approach involves an extensive evaluation through an experiment of the framework in use in the digital content domain by a group of domain experts in a controlled environment. The prototype is in operation, supporting and providing a platform where non-technical domain experts can customize their process model and configure the operational part of the process model through rules and models. We summarize the evaluation of the prototype of the framework in terms of its usability. The focus to evaluate the usability of artefacts is twofold: 1. Effectiveness and Efficiency of rule configuration 2. Satisfaction of overall framework in DCT domain. We review the principal findings and results of the research evaluation, summarizing the usability of the framework to an adaptation of configured rule in the process model customization. More specifically, we evaluate the usability of the framework for non-technical domain experts with specific claims:

- *Overall satisfaction:* Results from SUS and data analysis show that participants' opinions are positive with regards satisfaction and effectiveness of overall framework.

For both digital content technology and machine translation systems, users require more autonomic functionality. We consider the rule generation and configuration techniques for process model customization in the DCT domain that can achieve similar results using other techniques such as Service Oriented Architecture or Method Engineering or Service-as-a-service as described here.

Conclusions and Future Scope

The proposed rule generation and configuration approach are based on the given requirement of the end user. Based on previous activities of end user and work patterns, the feature models provide the recommendation of a feature at the time of feature selection. Additionally, from the generated rule, we describe the selected feature and vice versa. The approaches can be utilized in mining the configured rule, which can be applied for customization of process models. We recommend future steps during rule generation and configuration of the case information.

We examine the existing challenges of the evaluation strategies with usability criteria. Flexible usability evaluation criteria are explored to address the challenges of end-users and business strategists who wish to adapt the domain-specific rule (DSR) languages in process design, development and implementation. Exploring new evaluation strategies adaptable to business applications and coexistence of computational intelligence with DSR are the foci of the research. The experimental setup designed remotely for user queries is useful for appraising the DSR efficiency and performance. Empirical research is done evaluating qualitatively and quantitatively different configurations of DSR, comprehending the evaluation properties such as efficiency, effectiveness, interoperability and satisfaction as appropriate usability criteria. Both manual and semi-automatic configurations are tested, and semi-automatic mode of configuration appears more efficient and satisfactory providing DSR articulations with better performance, quality, learnability, user-friendly and reliability as sub-criteria. Semi-automatic mode of configuration appears furnishing better performance, efficiency and user satisfaction. For the approach to be generalizable i.e. to make applicable in multiple domains, we realize the need of further research, aiming at adapting the DSR across multiple domains and how the conceptual models are convertible into generic DSR languages. So far, the translation is semi-automatic, but can be improved with a system that learns from existing rules and domain models, driven by the feature approach, and results in an automated DSR generation.

Acknowledgement

This research is supported by Science Foundation Ireland (SFI) as a part of the ADAPT Centre at Dublin City University (Grant No: 12/CE/I2267).

References

Barišić, A., Amaral, V., Goulão, M., and Barroca, B. 2012. "Evaluating the Usability of Domain-Specific Languages", in M. Mernik, (Ed.): "Formal and Practical Aspects of Domain-Specific Languages: Recent Developments", *IGI Global*, 2012.

Bellamy, R, B. John, J. Richards, and J. Thomas, 2010. "Using CogTool to model programming tasks". Proc. Evaluation and Usability of Programming Languages and Tools (*PLATEAU 2010*), 2010.

Barišić, A., Vasco Amaral, V., Goulão, M. and Barroca, B. 2010. "Quality in Use of Domain-Specific Language: a Case Study", In Proceedings of the Workshop on Evaluation and Usability of Programming Languages and Tools (*PLATEAU 2011*) at SPLASH 2011, Portland, Oregon, USA, ACM, October 2011.

Barišić, A., Amaral, V., Goulão, M., and Barroca, B. 2011. "Quality in Use of DSLs: Current Evaluation Methods", In Proceedings of the *INFORUM'2011*, Coimbra, Portugal, September, 2011.

Barisic, A. 2017. Usability Evaluation of domain specific languages supported by USE-ME Framework, https://zenodo.org/record/345941/files/USE-MEFrameworkGuidelines1.1.pdf, https://doi.org/10.1145/3135932.3135953, 2017.

Lazar, J., J.H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. 2010: John Wiley & Sons.

Papulova, E. and Papulova, Z. 2006. "Competitive strategy and competitive advantages of small and midsized manufacturing enterprises in Slovakia," *E-Leader, Slovakia*.

I. 9241-11, "Ergonomic requirements for office work with visual display terminals (VDTs)–Part 11," *International Organization for Standardization,Geneva,* 1998 (it is updated as 9241-11: 2018).

Dix, A. 2009. *Human-computer interaction*: Springer.

Mani, N. Helfert, M. and Pahl, C. 2016. "Business Process Model Customisation using Domain-driven Controlled Variability Management and Rule Generation." *International Journal on Advances in Software* (Numbers 3 & 4, 2016): 179 - 190.

Tselios, N. Avouris, N. and Komis, V. 2008. "The effective combination of hybrid usability methods in evaluating educational applications of ICT: Issues and challenges," *Education and Information Technologies*, vol. 13, pp. 55-76, 2008.

Sauro, J. 2011. "Measuring Usability with the System Usability Scale (SUS)".

Kaplan, B. and Duchon, D. 1988. "Combining qualitative and quantitative methods in information systems research: a case study," *MIS quarterly*, pp. 571-586.

Mani, N. Helfert, M. and Pahl, C. 2017. A Framework for Generating Domain-specific Rule for Process Model Customisation. International Conference on Computer-Human Interaction Research and Applications (CHIRA), Funchal, Maderia- Portugal.

Basili, V.R. 1996. "The role of experimentation in software engineering: past, current, and future," in *Proceedings of the 18th international conference on Software engineering*, 1996, pp. 442-449.

Mani N., Helfert M., Pahl C., Nimmagadda S.L., Vasant P. (2018). Domain Model Definition for Domain-Specific Rule Generation Using Variability Model. In: Zelinka I., Vasant P., Duy V., Dao T. (eds) Innovative Computing, Optimization and Its Applications. Studies in Computational Intelligence, vol 741. Springer, Cham, https://doi.org/10.1007/978-3-319-66984-7_3.

Lewis, J.R. 1995. "IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use," *International Journal of Human-Computer Interaction,* vol. 7, pp. 57-78, 1995.

Bangor, A. P. Kortum, T. and Miller, J.T. 2008. "An empirical evaluation of the system usability scale," *Intl. Journal of Human–Computer Interaction,* vol. 24, pp. 574-594.

Molich, R. and J. Nielsen, J. 1990. "Improving a human-computer dialogue," *Communications of the ACM,* vol. 33, pp. 338-348.

Mani, N. and C. Pahl (2015). Controlled Variability Management for Business Process Model Constraints. ICSEA 2015, The Tenth International Conference on Software Engineering Advances, IARIA XPS Press.

Nielsen, J. 1994. "Heuristic evaluation," *Usability inspection methods*, vol. 17, pp. 25-62.

Tognazzini, B. 1992. *TOG on Interface*: Addison-Wesley Longman Publishing Co., Inc., 1992.

Preece, J. and Rombach, H.D. 1994. "A taxonomy for combining software engineering and human-computer interaction measurement approaches: towards a common framework," *International journal of human-computer studies,* vol. 41, pp. 553-583.