Evaluation of BehaviorMap: a User-Centered Behavior Language

Fernando Wanderley, Antonio Silva
NOVA-LINCS, FCT
Universidade Nova de Lisboa
Caparica, Lisboa
{f.wanderley, ap.silva}@fct.unl.pt

João Araújo
NOVA-LINCS, FCT
Universidade Nova de Lisboa
Caparica, Lisboa
joao.araujo@fct.unl.pt

Abstract**—**In the software development process, one of the recurring problems is to ensure that the expectations of stakeholders are being met. These expectations must match the system's behavior and be present in the requirements specifications and models. The Requirements Engineering discipline studies how to capture, specify, validate and manage requirements. However, recent empirical studies show that stakeholders do not usually understand traditional requirements models. This paper focuses on the cognitive evaluation of a user-centered language called BehaviorMap that aims to specify behavioral user scenarios in a cognitive way, based on mind map modelling. This paper describes an experimental evaluation to verify the understandability of the BehaviorMap scenarios compared to the textual ones. The experiment gathered data from 15 individuals (naïve-users), with different backgrounds, that had to analyze 8 scenarios, being 4 graphical and 4 textual. To assess the participants' cognitive effort, it was used questionnaires. Also, the time effort to perform the tasks was measured. This experiment showed promising results for the BehaviorMap scenarios**.**

Keywords—Agile Requirements; Mind Map Modelling; Behavior-Driven Design; User-Centred Requirements; Cognitive Effort.

I. INTRODUCTION

During software development, one of the frequent problems is to guarantee that the expectations of stakeholders are documented as requirements and must match the system's behavior. The Requirements Engineering (RE) discipline studies how to capture, specify, validate and manage requirements, which are normally represented using text or diagrammatic models. However, recent empirical studies [5] show that stakeholders do not usually understand commonly used requirements models to express system behavior (e.g., sequence and activity diagrams). With the introduction of agile methodologies, which have an ongoing collaboration of stakeholders during the development of software, the assumptions of the RE discipline were questioned and gave rise to new practices. One of those practices is Behavior-Driven Development (BDD), which emerged with the purpose of giving stakeholders the ability to express, in textual form, the behavior they desire for their software.

Nevertheless, it is well-known that the use of natural language to specify requirements can convey ambiguities and loss of information when the development team reads the behavior specifications provided.

To address these issues we designed the diagrammatic language BehaviorMap [21], a domain specific language (DSL) to improve cognitive aspects of BDD, through the cognitive properties of a Mind Map. The language belongs to a framework called SnapMind [21]. The SnapMind framework is composed of tools to specify both domain and behavioral models. The main premise of this framework is the use of mind maps, a cognitive model, to help increase the stakeholders' engagement in software development.

Our hypothesis is that by using mind map in requirements models, since it is a user-centered diagram, stakeholders will understand requirements more easily and consequently participate more productively during software development. To test our hypothesis, we produced an experimental evaluation to assess the cognitive effort of understanding BehaviorMap's and textual scenarios. Therefore, this paper reports the evaluation of BehaviorMap, a graphical DSL to define BDD scenarios. The experiment assesses the cognitive effort on understanding textual and graphical BDD scenarios. This experiment wants to verify if stakeholders can understand mind maps more easily comparing to textual written scenarios.

This paper reports the process used, our findings and problems that we encountered. We used questionnaires with questions about the scenarios to measure the cognitive effort. Time effort was also measured.

The document is organized as follows: section 2 presents the main characteristics of the SnapMind framework addressed in this experiment. Section 3 reports all the details regarding the measurements used, process, participants and results of the ex-periment. Section 4 reports related work of similar experiments and the final section draws some conclusions from this experiment and points out directions for future work

II. BACKGROUND

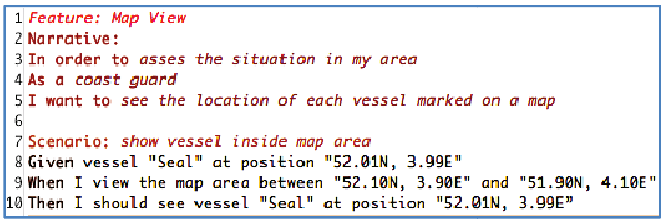
A. *Behaviour-Driven Development*

Behavior Driven Development is an agile practice to capture and test requirements, which evolved through the synergies between Test-Driven Development (TDD) and Acceptance Test-Driven Development (ATDD) [20]. In BDD, the main issue is behavior rather than testing. In BDD, behavior (which gives rise to tests) is specified directly with stakeholders through BDD scenarios. Thus, BDD evolves from ATDD with the objective of obtaining the description of system behavior, taking advantage of an active communication with stakeholders. With the information obtained, acceptance tests to verify the expected behavior are created. BDD arises with the aim of incorporating stakeholders in the development process, improving testing-

978-1-4673-6630-4/15/$31.00 ©2015 IEEE

oriented practices. It creates an abstraction for stakeholders to be part of the development process. So, BDD is characterized by being a requirements practice and, at the same, time, an acceptance tests construction. BDD extends user stories to develop scenarios. A scenario is a possible behavior that is expected to occur from a user story. Each user story is instantiated with multiple BDD scenarios, where each one is the outlined expected behavior for that situation. The scenario is built using the domain language of the project and is written by stakeholders in conjunction with the development team. BDD scenarios have a template to be used in any situation. JBehave is the current tool used to specify scenarios in BDD context.

Figure 1 shows a BDD scenario written in JBehave. The scenario describes the situation that allows a coast guard to see a vessel given a location in a map; this information must be used to assess the situation in an area supervised by the coast guard.



```
Feature: Map View
Narrative:
In order to asses the situation in my area
As a coast guard
I want to see the location of each vessel marked on a map

Scenario: show vessel inside map area
Given vessel "Seal" at position "52.01N, 3.99E"
When I view the map area between "52.10N, 3.90E" and "51.90N, 4.10E"
Then I should see vessel "Seal" at position "52.01N, 3.99E"
```

Fig. 1. User Scenario in JBehave approach.

B. *Mind Maps*

A mind map is a diagram used to view, classify and organize concepts, and to generate new ideas in a simple and intuitive way. It is used to connect words, ideas and concepts to a central idea or concept [4]. It is similar to a semantic network, or a cognitive map, but without restrictions on the types of connections used. A mind map is a radial diagram that, through a vocabulary (i.e., set of keywords), may cognitively represent and model a concept or a specific domain. According to Buzan [4], the main benefits of using mind maps are: organization of ideas and concepts, emphasis on the relevant keywords, association between elements in branches, grouping ideas, support visual memory and creativity, and trigger innovation. Both the academy and the industry consider several techniques and tools based on mind maps powerful tools for managing the process of eliciting requirements in agile development [25, 26, 27, 28]. Mahmud [25] shows interesting results collected during an experiment with experts and non-experts in mind maps. Other authors [29, 30] introduce the idea of using mind maps as one of the agile practices to obtain and represent requirements

These diagrams are the bases of our approach for requirements models.

C. *SnapMind Framework*

The SnapMind framework aims to make the requirements modeling process more user-centered, through the definition of a visual requirements language, based on mind maps, model-driven and domain specific language techniques. Also, through these techniques, the SnapMind framework focuses on the support for consistency between user stories and the domain models using a snapshots technique [21]. The idea is to improve the cognitive effectiveness and involvement of domain experts and business specialists (as collaborative modelling resulting in a feedback mechanism). The BehaviorMap language, one of SnapMind's languages, was the one evaluated by the experiment reported in this paper.

The SnapMind framework is mainly composed of three components: the Domain Modelling Visual Editor, the User Scenario Visual Editor, and the USE-tool. Both editors are key to make the requirements modelling process more user-centred. They are used to build domain model and user scenario both based on mind maps. The idea is to improve the cognitive effectiveness and involvement of domain experts and business specialists (as collaborative modelling resulting in a feedback mechanism).

As shown in Figure 2, the editors transform the models into USE input artifacts format aiming to check inconsistences between them.

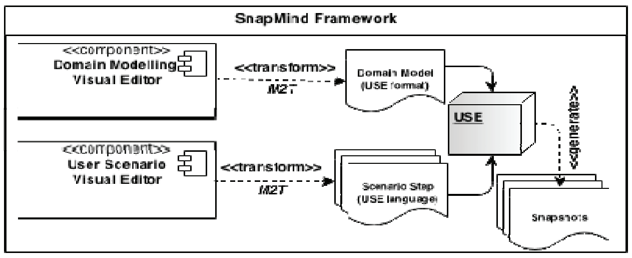

SnapMind Framework
<<component>>
Domain Modelling Visual Editor
<<transform>>
M2T
Domain Model (USE format)
USE
<<generate>>
<<component>>
User Scenario Visual Editor
<<transform>>
M2T
Scenario Step (USE language)
Snapshots

Fig. 2. The SnapMind Framework.

The tool receives as input: the domain model in USE format and; a set of USE script commands, each one related to a Given, When or Then step of the user scenario represented by a mind map. The USE tool is used as a black box, i.e., it serves as a filmstrip generator – a filmstrip is a sequence of snapshots-which serves as a mechanism for verification and validation models for developers. In this paper, we will focus on the scenario module of the framework.

To use the framework we defined a process as shown in Figure 3. It consists of three main activities.

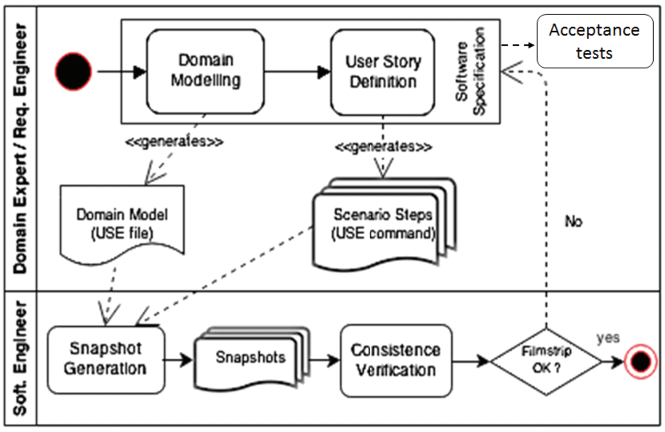


Fig. 3. The SnapMind process.

The first activity is the software specification that includes both the definition of the domain model and the user story. The second activity is concerned with the validation models through animation, involving snapshots generation, executed by the USE tool. The third activity is the consistence verification activity, reported by USE.

The Domain Modelling phase of the Specification activity is an iterative and incremental process. First the requirements engineer captures the essential elements of the system domain together with the domain expert or business specialist and after that, the requirements engineer refines this conceptual mind map adding specific concerns.

The User Stories definition is also a sub activity of software specification where the stakeholders write the expected behaviors from the system in natural language, as shown in Figure 1. After the user stories are specified, the requirements engineer represents these scenarios using the visual language editor and show them to the end-user, for validation purposes, where user feedback is expected.

So, the domain modelling and user stories specification activities only generate their output artifacts (for validation) after the end-user feedback. These artifacts are then transformed into USE text files (conceptual model and scenarios) to be used in snapshots generation. Finally, the Software Engineer will verify the consistency between requirements models, reported by USE.

A set of transformation rules are defined using the mind map metamodel, the USE domain model and the USE BNF grammar. The transformation language EGL was used to implement these rules. Moreover, the SnapMind framework also provides static analysis through the USE-tool, as a separated concern of developer's team.

Additionally, acceptance tests specifications are generated automatically from the scenarios specified using BehaviorMap. These tests specifications are derived using the information provided by Given, When and Then steps of the scenarios.

The User Story Visual Editor. The abstract syntax definition extends the mind map metamodel using inheritance strategy. We can observe in the metamodel, shown in Figure 4, that the main elements to represent a user story (in an acceptance test) are the set elements {Step (Given, When, Then), Entity, Attribute, Value}, which extend the main elements of the mind map metamodel.

For example, the abstract element Step <<extends>> Node element from the mind map metamodel. Each Step defined contains many Entities; each Entity is composed by many Attributes where each Attribute has one and only one Value. The metamodel implies the order and cardinality of Steps. The first Step is Given, followed by When and finally by the Then step node. This order constraint is reinforced with rules. In this sense, well-formed rules were defined to improve the semantics of our visual language for the User Story Editor in EVL. This rule specifies the implication of the order of Steps defined for a User Story, which will always have a Given, When and Then (in this order). Eventually, it will be necessary to create another scenario, if the user story may need another When and Then steps for the same Given, as the scenario shown in Figure 1. In SnapMind, user stories represent a specific state of the system at a given time (t). For this reason, the actual stage of User Scenario Visual Editor just represent only Given, When and Then steps.

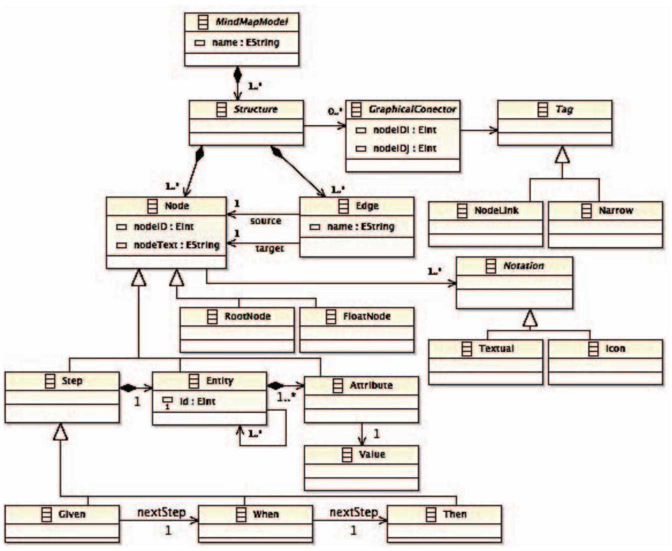


Fig. 4. Metamodel for User Story based on Mind Map.

As a result of the metamodel definition with EVL rules and the concrete syntax, a model built using the visual editor for User Story has the appearance as shown in Figure 5. This is a visual representation for a user story depicted in Figure 1. The scenario shows an example where, provided that the vessel "Seal" is in a given position, a coast guard, by visualizing a map area limited by start and end points, will see this vessel inside this area.

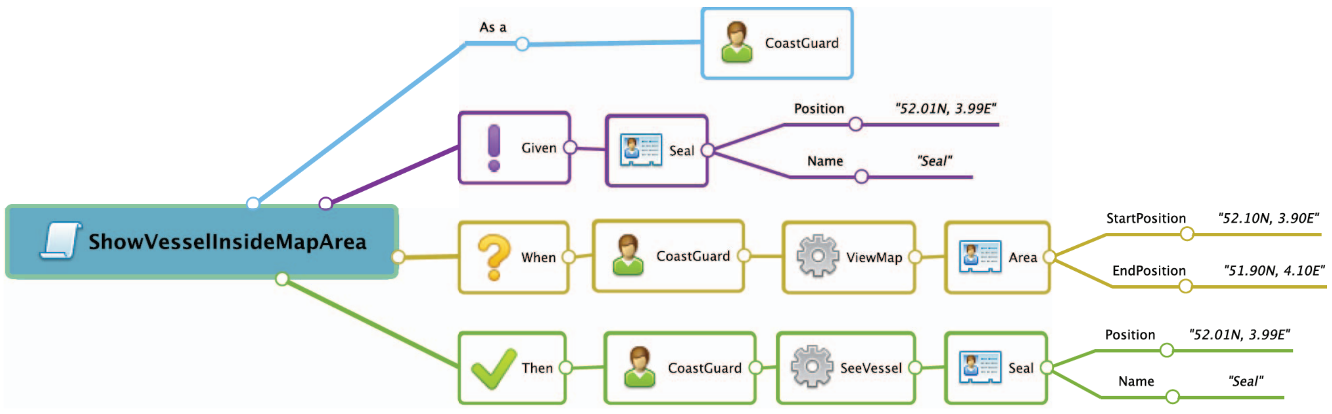


Fig. 5. BehaviorMap's example scenario – Show vessel inside map area.

III. EXPERIMENTAL EVALUATION

The understanding of BehaviorMap's scenarios by ordinary users was the target of the evaluation performed. From this assessment we wanted to find out if users had a lower cognitive effort when confronted with diagrams specified by BehaviorMap in relation to textual scenarios. Cognitive effort is a theory, which states that the capacity of the working memory of a person is limited and must be used effectively [17]. The total load of cognitive effort is a sum of the *Intrinsic* effort, i.e., the difficulty and complexity of information), the *Germane* effort, with regard to the effort of learning and memorization and the *Extraneous* effort, related to how information is presented.

A secondary concern of the evaluation was the test coverage provided by the BehaviorMap tool in comparison to traditional textual ones. This metric was measured by comparing the total assertions generated by the BehaviorMap tool versus the textual BDD ones. The processes consisted in collecting textual scenarios from various sources and translate them into BehaviorMap scenarios to verify for the same scenario the test coverage produced by the two approaches. Section H details how this process was performed.

A. Experiment Design

The experiment was designed to test the hypothesis whether BehaviorMap scenarios are more easily understood than textual scenarios. The experiment consisted of: a training task, two practical tasks and six comprehension tasks. Each participant performed the experiment individually, accompanied by one of the paper's authors.

1) Training Task

The training task lasting 30 minutes maximum, aimed to explain to the participants the elements of the experiment. Initially, it was explained to the participant that the scenarios were used to represent behaviour and address two different types: textual and graphical. Then, it was shown two examples of scenarios; written textually and graphically; in order to the participant check the similarities and differences between them. When explaining the graphical scenarios, a description of what each node meant and how to read a scenario was given to the participant. After that, the tools, JBehave (the current tool used by industry) and BehaviorMap, were explained. Finally, the participant did the examples scenarios using the respective tool and the help facility when needed.

2) Practical Tasks

The practical tasks were designed to confront participants with BehaviorMap and JBehave tools. They were always performed before the comprehension tasks for participants to interact with the tools. In practical tasks the participants had to make a translation from a textual scenario to a graphical scenario and vice versa. The participants had seven minutes in total to conduct the task.

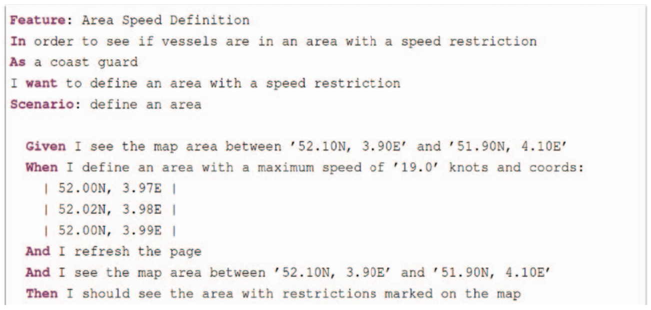
3) Comprehension Tasks

The comprehension tasks served to assess how participants understood the graphical and textual models by answering questions about them. Participants were exposed to the scenario and three questions (exposed at the top right of the screen). The participants then had to answer these three questions within five minutes. The questions used were:

- What are the initial conditions expected?
- What are the actions to be specified?
- What is the expected result?

B. Selected Scenarios

The selected scenarios used in the experiment were collected from multiple BDD sources [9, 10, 13–15] and translated to BehaviorMap language. The scenario shown in Figures 1 and 5 was one of the selected scenarios in the experiment. Its complexity was classified as medium. An example of a scenario of high complexity is given in Figures 6 and 7.



```
Feature: Area Speed Definition
In order to see if vessels are in an area with a speed restriction
As a coast guard
I want to define an area with a speed restriction
Scenario: define an area

  Given I see the map area between '52.10N, 3.90E' and '51.90N, 4.10E'
  When I define an area with a maximum speed of '19.0' knots and coords:
    | 52.00N, 3.97E |
    | 52.02N, 3.98E |
    | 52.00N, 3.99E |
  And I refresh the page
  And I see the map area between '52.10N, 3.90E' and '51.90N, 4.10E'
  Then I should see the area with restrictions marked on the map
```

Fig. 6. Textual version – Speed information of non-anchored vessel.

This scenario specifies the case where the coast guard needs to check whether the vessels in a determined area are compliant to the speed limit. The BehaviorMap version is shown in Figure 7.

From that set of scenarios a set of metrics was applied. The metrics were designed to try to differentiate multiple BehaviorMap's scenarios. The metrics conceived were: (i) Scenario Size to count the leafs in scenario branches (Given, When, Then); (ii) Actions in *When* branch (*ActionsWhen*); (iii) Actions in *Then* branch (*ActionsThen*) and (iv) Distinct Entities (*Entities*). These metrics are similar to those found for UML diagrams, such as the ones encountered for class, sequence or activity diagrams [19]. By applying the metrics, 8 scenarios were selected: 4 textual and 4 graphical.

The scenarios were grouped into three complexity levels: low, medium and high. For the high complexity level, the scenarios chosen were those that covered more sets of maximum values of all metrics applied. For the low complexity level, the chosen scenarios were those with the minimum values and for the middle level the scenarios that had average values. Table I shows the metrics values of the selected scenarios.

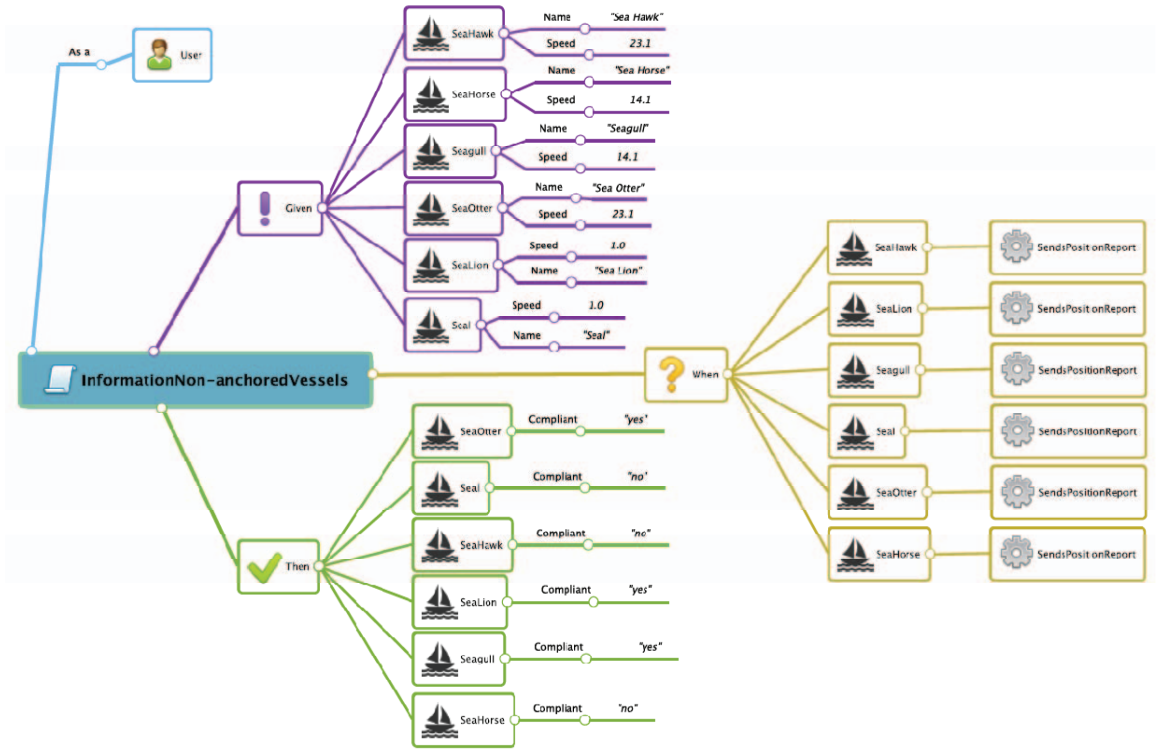


Fig. 7. High complexity scenario – Speed information of non-anchored vessel.

TABLE I. SCENARIO' METRICS OF THE SELECTED SCENARIOS.

Scenario Description	*Entities*	*Scenario Size*	*Actions When*	*Actions Then*
Textual High	8	28	1	1
Graphical High	6	24	6	0
Textual Medium	2	6	1	1
Graphical Medium	2	6	1	1
Textual Low	1	3	1	0
Graphical Low	1	3	1	0
Training BM	1	4	1	0
Training Textual	2	5	1	2

C. *Measurements*

The experiment had several forms of information capture, used in different situations or in parallel. The sources of information used were questionnaires and answers, and time effort measurement.

1) ***Questionnaires***

Three questionnaires were used. The first questionnaire was used in order to obtain information on the profile of participant, i.e., his name, age, background and if had some experience with behavioural models and BDD scenarios.

The second questionnaire was used to capture the cognitive effort of the participants after the completion of a task. The questionnaire used was the NASA-TLX, a questionnaire quite used to subjectively assess the cognitive effort of a person performing tasks [12]. The questionnaire was delivered to the participants after the completion of a task. He made a self-evaluation concerning the attributes of performance, mental effort, temporal effort physical effort, level of frustration, etc. Each attribute had a different weight.

The third and final questionnaire served to make an assessment of the whole experiment by the participant, i.e., if he liked it, if he found that the information given in the training session was appropriate, his classification on the difficulty of each task on a scale from 1 to 10 and the possibility of giving a written opinion.

2) *Temporal Effort and Answers*

To assess the difficulty of the participants to perform the tasks, time effort was measured, and their answers evaluated. Time was initiated with a stopwatch when a participant began the task and ended when he/she finished. The answers in the case of comprehension tasks were recorded with an audio recorder so they can be analysed later.

D. *Subjects*

The subjects in this experiment were selected taking into consideration different backgrounds. This decision was taken because the diagrams proposed to represent the behavioural scenarios were constructed not to focus on any in particular background. The only restriction to participate in the experiment was if the participant mastered the English language. In total 15 people participated in the experiment, nine males and six females. Four of the participants had no area of expertise. Three of them belonged computer science and two of them to electronic engineering. Four participants are from nature sciences (Veterinary Medicine, Nutrition and Biology). One belonged to the area of social communication.

E. *Work Environment*

The experiment was conducted in various locations, depending on participant's availability. All experiments were performed on the same computer, a laptop using a wireless mouse. The distance of the laptop was adjusted depending on the participant, until it was visually pleasant to him.

F. *Experimental Process*

In the first activity (***Activity 1***), it was explained to the participant how he would run the experiment and what his role in it was. Also he had to fill in a questionnaire about his background. The second task, served to given a small lecture to the participant about the concepts of BDD. After this introductory part, the practical tasks were started. Before the beginning of the tasks, it was explained to the participant that the goal of practical tasks would be to perform a translation from a textual scenario to a graph and vice versa (***Activity 2***). It was also introduced the NASA-TLX questionnaire that the participant had to answer at the end of each task. Then, the participant performed the practical task (***Activity 3***). During the execution it was started a stopwatch to count the duration of the task.

After finishing the task, the timer was stopped and the participant responded to the NASA-TLX. Any questions raised by the participant on the questionnaire was clear and had unlimited time to respond to it. If there were more practical tasks to execute, the process repeats itself from Activity 3. Otherwise, the process followed for the comprehension tasks (***Activity 4***).

The comprehension tasks were performed in random order of complexity, i.e. when performing a textual task of lower complexity the next task would be with the same complexity but with a graphical scenario. If any answer from a participant were considered incomprehensible, it would be asked to repeat it. The participant performed the task and then responded to the NASA-TLX questionnaire (***Activity 5***). These activities were repeated for each task of scenario understanding. After all comprehension tasks being performed, the time was registered and the participant responded to the final questionnaire (***Activity 6***). After answering the questionnaire, questions were posed to the participant about what he thought about the experiment and what type of scenarios he preferred.

G. *Results*

The data were compared using a nonparametric analysis of variance using the Kruskal-Wallis [23] and Mann-Whitney [24] methods. The choice of nonparametric analysis was due to the fact that the data do not have a normal distribution. The Anderson-Darling test was performed and it was concluded that the data did not have a normal distribution with a confidence level of 99%. Analyses of variance were performed to consider two factors, the way the BDD scene was written (textually or graphically) and its complexity level to see if there are significant differences with changing complexity. All results were obtained with a confidence level of 95%.

1) *Practical Tasks*

In practical tasks, the goal was to make a translation from textual to chart and vice versa. Table II presents the average results for each characteristic (workload, time and correct answers) in relation to practical tasks. The TP1 task (translation from a mind map to text) averaged workload is lower compared to the TP2 (translation of text to mind map), a time duration on average lowered by 2 minutes. In relation to positive responses, we got 90% on average for the TP1 task and 50% for the TP2 task, i.e., a 40% difference.

TABLE II. SUMMARY OF RESULTS FOR PRACTICAL TASKS.

Measurement	***TP1***		***TP2***	
	x	**s**	**x**	**S**
Workload (0-100)	26.26	18.17	50.70	25.00
Temporal Effort (m)	4.29	1.31	6.11	1.13
Correct Answers (%)	90.77	17.00	50.77	21.77

Analyses of variance of the practical tasks assume that variation is due to the original scenario that is offered and not the use of the tool used. Participants performed easier the TP1 task compared to TP2 according to the runtime, the workload and the percentage of correct translations. In relation to the workload, most of the results (75%) of the TP1 task have values up to 35 and the TP2 task has values up to 70. Regarding the time of realization of task, 75% of the times are below 5 minutes for the task at TP1 compared to 7 minutes (time limit) for the task TP2.

Finally, regarding the compliance of the translation done, TP1 task for the remaining participants have 92% or 100% of the translation note. In the TP2 task, 75% of participants had grades below 60%, where there was almost a 100% performance by one participant. In order to verify whether the differences between the means were not caused by random events, an analysis of variance (Mann-Whitney) was applied. The result (U = 174.0, P = 0.01) showed that the average difference in workload between tasks was not caused by random events, but rather the difference between the types of scenarios. In relation to the time effort and translations it has also used an analysis of variance (Mann-Whitney). The analysis result showed that the difference between the values of time effort (U = 196.0, p = 0.00) and translations (U = 23.0, p = 0.00) were not caused by random factors, but due to the differences between the types of models.

2) *Comprehension Tasks*

In the comprehension tasks, the participants had to answer three questions for each model, in a time slot of five minutes. Six models were shown in total, three graphical and three textual. Each model presented belonged to a level of complexity (low, medium, high). Table III shows the means (x) and standard deviations (s) of the graphical and textual tasks. According to data recorded, tasks using graphics behavioral models had lower workload and time effort and got more correct answers. For all the measurements, two-variance analyses were performed. One analysis fixing the scenario type and varying the complexity class (AV-I), and other fixing the complexity class and varying the scenario type (AV-II). When describing the analysis, we use the codes AV-I and AV-II.

TABLE III. SUMMARY OF RESULTS FOR COMPREHENSION TASKS.

Measurement	*Graphic*		*Textual*	
	x	s	x	s
Workload (0-100)	24.68	21.75	34.54	28.17
Time (m)	1.64	1.18	2.09	1.29
Correct Answers (%)	96.77	16.30	84.36	31.53

3) *Congnitive Effort*

Figure 8 shows the average workload of each task. The AV-I analysis showed that the graphical scenarios (H = 12.48, p = 0.0019) had no significant difference between them according to the changing of complexities (confidence of 95%).

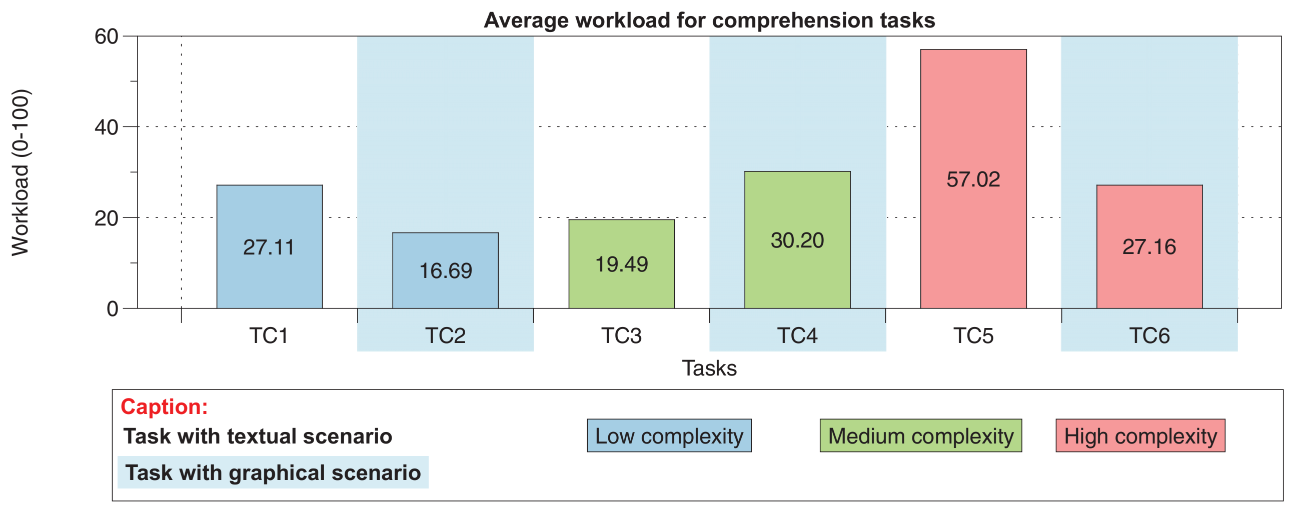


Fig. 8. Average workload for comprehension tasks.

Regarding textual scenarios (H = 3.25, *p* = 0.1969), the same did not happen. To determine which average differs in the analysis of workload a Mann-Whitney analysis was used. The results are shown in Table IV. The analysis concluded that there was a significant difference in the scenarios of low and medium complexity in relation to the high complexity scenarios, showing that workload increases with the high complexity textual scenarios.

TABLE IV. KRUSKALL-WALLIS POST HOC ANALYSIS WITH MANN-WHITNEY FOR TEXTUAL SCENARIOS.

Comparing	**U, *p***	**Conclusion (▯= 0.05)**
Low vs. Medium	133.0, *0.40*	No significant diff.
Medium vs. High	192.0, *0.00*	**Significant difference**
Low vs. High	49.0, *0.01*	**Significant difference**

In the AV-II analysis it was found that the differences recorded for the low and medium complexity scenarios were not significant. However, for high complexity scenarios the same did not happen. The analysis showed that for high complexity scenarios, the scenario type impacts the workload. The results are shown in Table V.

TABLE V. MANN-WHITNEY ANALYSIS – FIXING SCENARIO COMPLEXITY AND VARYING SCENARIO TYPE.

Comparing (scenario type)	**U, *p***	**Conclusion (▯= 0.05)**
Low	78.0, *0.15*	No significant difference
Medium	141.5, *0.23*	No significant difference
High	50.0, *0.01*	**The scenario type impacts the workload**

4) *Time Effort*

Figure 9 shows the mean values of recorded time effort for each task. Regarding the AV-I analysis, the results for textual scenarios (H = 18.88, p < 0.0001) and for graphical scenarios (H = 11.67, p = 0.0029) showed that the complexity influenced both of them. In order to determine the average time effort that differs in both types of scenarios, the Mann-Whitney analysis was applied.

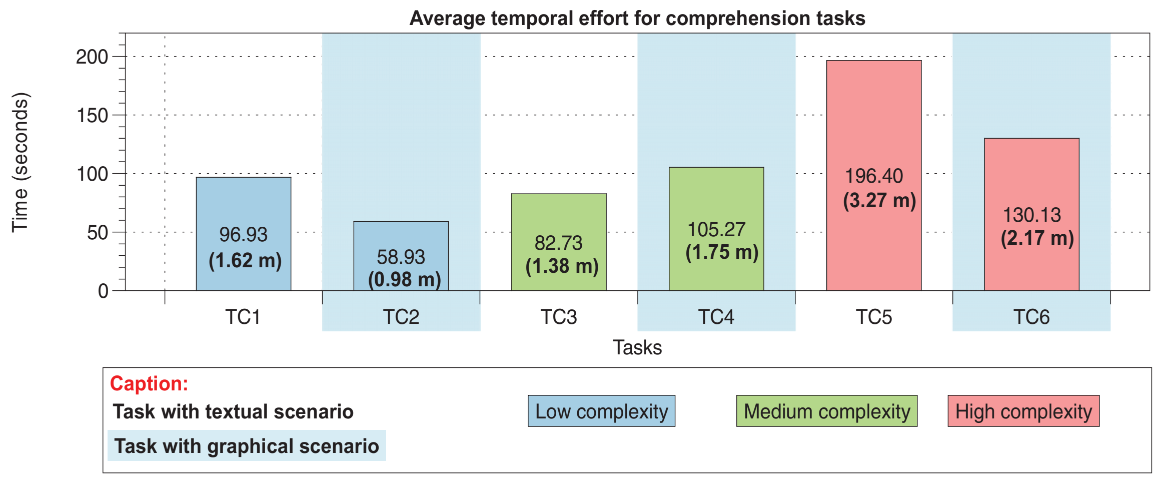


Fig. 9. Average temporal effort for each comprehension task.

Table VI shows calculations made for textual scenarios. We can conclude that the tasks with high complexity affected the time effort compared to the other two levels of complexity as expected. Table VII shows calculations made for graphical scenarios. The results show that low complexity level has significant lower values than the middle and high levels. In this case, this significant difference is advantageous, because indicates that the participants concluded the task very fast in relation to the others. Results are presented with 95% confidence level.

TABLE VI. KRUSKALL-WALLIS POST HOC ANALYSIS WITH MANN-WHITNEY FOR TEXTUAL SCENARIOS.

Comparing	**U, *p***	**Conclusion (α= 0.05)**
Low vs. Medium	105.5, *0.77*	No significant diff.
Medium vs. High	18.0, *0.00*	**Significant difference**
Low vs. High	198.0, *0.00*	**Significant difference**

TABLE VII. KRUSKALL-WALLIS POST HOC ANALYSIS WITH MANN-WHITNEY FOR GRAPHICAL SCENARIOS.

Comparing	**U, *p***	**Conclusion (α= 0.05)**
Low vs. Medium	162.5, *0.04*	**Significant difference**
Medium vs. High	145.5, *0.17*	No significant diff.
Low vs. High	193.0, *0.00*	**Significant difference**

In the AV-II analysis for the time effort, the results showed that the differences of means in the lower and higher complexity levels were not caused by random factors, but by the difference in the type of scenarios. Regarding the scenarios of medium complexity, the results showed that there were no significant differences. Table VIII presents the results obtained.

TABLE VIII. MANN-WHITNEY ANALYSIS – FIXING SCENARIO COMPLEXITY AND VARYING SCENARIO TYPE.

Comparing (scenario type)	**U, *p***	**Conclusion (α= 0.05)**
Low	65.0, *0.05*	**The scenario type impacts the temporal effort**
Medium	129.0, *0.49*	No significant difference
High	56.5, *0.02*	**The scenario type impacts the temporal effort**

5) *Correct Answers*

Figure 10 shows the average correct answers for the comprehension tasks. The AV-I analysis showed that the graphical scenarios (H = 6.65, p = 0.036) had no significant difference between them according to the changing of complexities. Regarding textual scenarios (H = 0.4, p = 0.8187), the same did not happen. In order to analyse which complexity level that generated the difference in the analysis of variance, the Mann-Whitney analysis was applied. Table IX shows calculations performed. We concluded that the high complexity level affected the differences in averages**.**

The AV-II analysis showed that there were no significant differences in low and medium complexity levels. For the high complexity level there were significant differences to affirm that type of scenario influenced the recorded responses. The analysis results are shown in Table X.

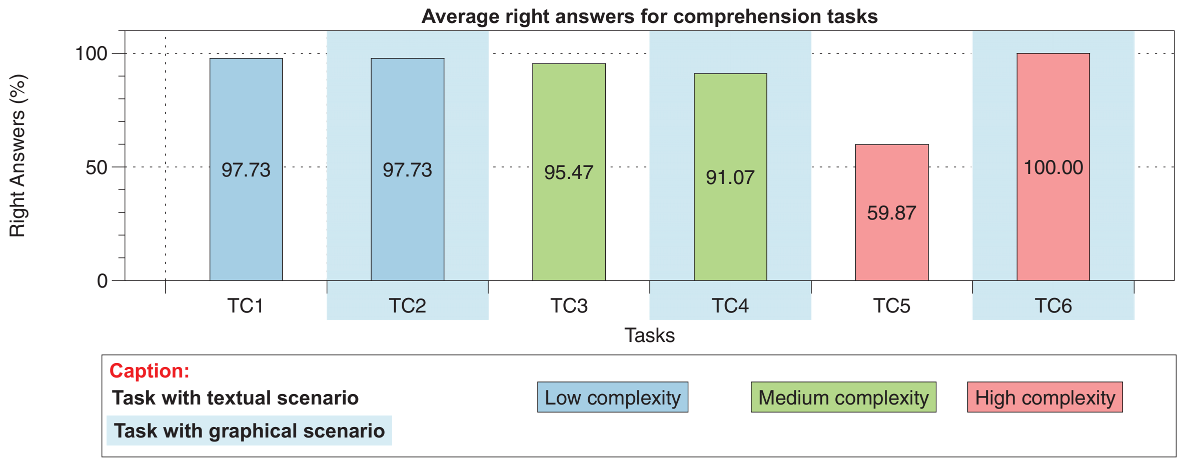


Fig. 10. Average right answers for each comprehension task.

TABLE IX. KRUSKALL-WALLIS POST HOC ANALYSIS WITH MANN-WHITNEY FOR TEXTUAL SCENARIOS.

Comparing	U, *p*	Conclusion (α= 0.05)
Low vs. Medium	105.0, *0.76*	No significant diff.
Medium vs. High	163.5, 0.03	**Significant difference**
Low vs. High	168.0, 0.02	**Significant difference**

TABLE X. MANN-WHITNEY ANALYSIS – FIXING SCENARIO COMPLEXITY AND VARYING SCENARIO TYPE.

Comparing (scenario type)	U, *p*	Conclusion (α= 0.05)
Low	112.5, 1.00	No significant difference
Medium	111.5, 0.97	No significant difference
High	172.5, 0.01	**The scenario type impacts the answers**

6) *Summary*

Through this experiment we can say that BehaviorMap showed better results in relation to the textual scenarios. The summary of results is presented in Table XI, Table XII and Table XIII.

TABLE XI. SUMMARY FOR PRACTICAL TASKS.

Measurement	*Result*
Workload	With significant differences (**better for BehaviorMap**)
Temporal Effort	With significant differences (**better for BehaviorMap**)
Correct Answers	With significant differences (**better for BehaviorMap**)

In practical tasks the results were clearly more favourable for the translation of text to BehaviorMap. The difficulties of the participants were recorded during the execution of the task and questioned after the experiment. The participants responded that they had difficulty in distinguishing what was an entity, attribute and value in the translation from text to mind map. Even those who achieved better results in translation of text to mind map took longer to get it. It should be noted that the use of BehaviorMap tool may have influenced the translation of the text to mind map in some cases, but the main reason told by the participants was the inability to distinguish the text elements.

Regarding the comprehension tasks, the analysis by setting the type of scenario and varying complexity levels, the graphical scenarios maintained a steady performance where there was only a significant lower difference in time effort for the low complexity level. However, this is advantageous as it demonstrates that the participants realized the scenarios more quickly. In textual scenarios, there have always been significant differences in the three measurements for the high complexity scenario. In the analysis where the complexity class was fixed, but varying the type of scenario, we found significant differences between the high complexity scenarios, where graphical scenarios did better. There was also a major difference for time effort, regarding the low complexity class, where the graphical scenario performed better.

TABLE XII. SUMMARY OF RESULTS FROM COMPREHENSION TASKS – FIXING SCENARIO TYPING AND CHANGING COMPLEXITY.

Measurement	*Scenario*	*Result*
Workload	Textual	More effort required in high complexity class than other classes
	Graphical	**No significant differences were recorded**
Temporal Effort	Textual	More effort required in high complexity class than other classes
	Graphical	**Less effort in low complexity class than other classes**
Correct Answers	Textual	More effort required in high complexity class than others
	Graphical	**No significant differences were recorded**

TABLE XIII. SUMMARY OF RESULTS FROM COMPREHENSION TASKS – FIXING COMPLEXITY AND CHANGING SCENARIO TYPE.

Measurement	*Complexity*	*Result*
Workload	Low	No significant differences
	Medium	No significant differences
	High	**With significant differences (Better for BehaviorMap)**
Temporal Effort	Low	**With significant differences (Better for BehaviorMap)**
	Medium	No significant differences
	High	**With significant differences (Better for BehaviorMap)**
Correct Answers	Low	No significant differences
	Medium	No significant differences
	High	**With significant differences (Better for BehaviorMap)**

H. Evaluation of Test Coverage

This part of the evaluation has the purpose to assess the test coverage provided by BehaviorMap scenarios versus traditional textual ones. In textual BDD approach, the information provided on tutorials and tools, says that the assertions are only specified based on the Then step, forgetting the information specified in step Given and When. The BehaviorMap approach creates assertions taken from the specifications of the Given, When and Then steps [21].

In order to verify this premise we collected a sample of 53 textual scenarios from multiple sources [9, 10, 13–15] and translate them to BehaviorMap scenarios. The translation rules consisted in finding nouns, verbs and adjective to represent the concepts in BehaviorMap. For the Given step a noun would become an entity and his adjectives attributes with a respective value. As for the When step a noun would become an entity and a verb will be considered an action. Finally, for the Then step, same as for the Given step, the nouns would become entities and his adjectives attributes with respective values. Furthermore if there were verbs associated with nouns it will generate an action for the respective entity. The BehaviorMap scenario showed at Fig. 5 results as a translation from the textual scenario presented at Fig. 1 using the process described.

After analyzing each scenario collected the total assertions were count. Fig. 11 shows a graph that shows the number of assertions generated for each of the approaches for the 53 collected scenarios. It shows that the BehaviorMap approach provides more test cases without increasing the effort time of the users, as the tests are automatically created.

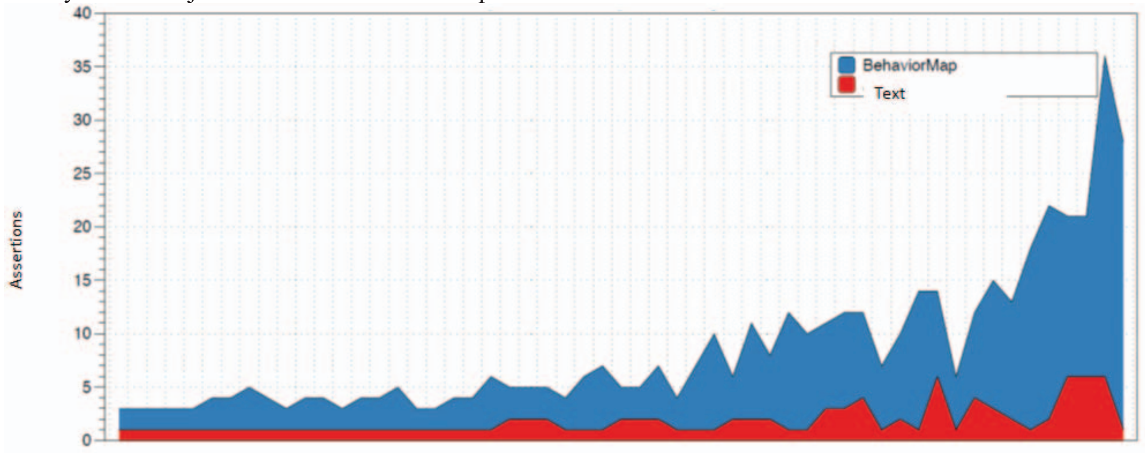


Fig. 11. Evaluation of test coverage.

IV. Discussion

This first experiment showed some evidence that BehaviorMap scenarios are easier to understand in relation to textual scenarios, especially when considering scenarios with higher complexity. Nevertheless, more experiments are needed with more scenarios and people with different backgrounds. The experiment stated metric evaluation of BDD scenarios proposed by the authors of the paper, these metrics may also require modifications to better analyse scenarios. However, the scenarios chosen for this experiment were able to cover different types of scenarios, showing that textual scenarios are more complicated to understand in relation to diagrams.

Namely, in practical tasks, the results showed it was clearer and easier to translate correctly a graphical scenario to text than a textual translation to graphical. The analysis admitted that the participants knew the BehaviorMap's DSL and had difficulty only in translation. This assumption is not completely correct, despite the training offered at the beginning of the experiment. However, when confronting participants in relation to the task, they mentioned the fact that they could not identify easily the text elements in the scenario to make the translation.

Regarding the results, the BehaviorMap had better results with the increasing of complexities of the scenarios, however, the textual scenarios had good performances in low and medium complexity levels. Even experienced difficulty in scenarios of high complexity, an average of 60% correct answers was obtained. In this initial experiment, there are some indications that there are certain cases in which information presented graphically has advantages over text, but to reinforce this claim, more experiments need to be conducted. It is noteworthy that the BehaviorMap applies to small and medium projects, which adopted agile processes in their software lifecycle development. The BehaviorMap is a special cognitive support to Behavior Driven Development in the scenarios specification addressing the end-user understanding.

V. Related Work

Evaluation of cognitive effort is a commune practice in engineering. Several works have been done to assess this effort. They have mostly used subjective measurements like the ones use in this experiment. However, there has also been reported the use of biometric sensors to better assess cognitive effort. The use of the NASA-TLX questionnaire has been widely used in different tasks for cognitive effort [11]. Hart's work reported a survey, with 550 studies, that the questionnaire was used, for a variety of tasks (e.g. visual displays; vocal/manual input devices; virtual or augmented vision). Regarding software development, the NASA-TLX has been applied to evaluate tools for collaborative source code development [7], collaborative user systems [16] and to test scrolling interfaces [6] and natural languages [18].

Rastkar's work [18] explored the creation of natural language summary for crosscutting source code concerns to help software change tasks. From source code, documentation is created to help programmers make decisions about the relevance of a concern to a change task being performed. To evaluate if the summaries help programmers do change tasks, an experiment was performed where participants had to make changes in small software with and without concern summaries. To assess the participants' effort, the NASA-TLX questionnaire was used. They concluded that the participants performed the tasks more easily with the help of summaries.

The use of sensors to better infer cognitive effort it is also become regular in the engineering's fields. Several studies regarding air flights [22], air traffic control [3], car driver status in different conditions [2] have use EEG equipment assess more precisely the participants effort in combination with the NASA-TLX questionnaire. In software engineering, cognitive effort studies using biometric sensors mainly focuses on eye tracking measurements. For example, Crosby's [8] study was to scan patterns and strategies of high and low experience developers. This work showed that high experienced developers spent less time reading comments than lower experienced ones. Begel's [1] experiment used sensors to assess the cognitive effort of understand C# code. They used three sensors (EEG, EyeTracking and ECG) to understand what sensors gave the best measurement alone and what the combination that got the best result was. They concluded that the three sensors combined gave the better result.

VI. Conclusion

In this work we contribute with an initial assessment to verify whether stakeholders understand more easily BehaviorMap's diagrams in relation to textual scenarios. With our proposed metrics we showed that BehaviorMap's scenarios had more consisted results considering the different complexity levels in comparison to textual ones. In the high complexity levels, textual scenarios got significant worse results compared to the other two complexity levels. This did not happen for BehaviorMap's scenarios. Also when comparing the results only considering the high complexity level, the graphical scenarios had significant better results than the textual ones. As a clear future work is repeat this evaluation exploring user-centred usability strategies for instance interviews and 'think-a-loud' strategies [31].

Nevertheless, replication of the experiment to substantiate this assessment is needed, with a larger number of participants and other scenarios. One point of improvement in the experiment will be the use of biometric sensors to enhance the cognitive effort measurement precision.

Acknowledgment

The authors would like to acknowledge - CNPq – Ciências sem Fronteiras, and NOVA-LINCS/FCT/UNL for the financial support for this work.

References

[1] Begel, A., Fritz, T., Mueller, S., Elliott, S., Zueger, M.: Using Psycho-Physiological Measures to Assess Task Difficulty in Software Development. Proceedings of the International Conference on Software Engineering. International Conference on Software Engineering (2014).

[2] Brookhuis, K.A., de Waard, D.: The use of psychophysiology to assess driver status. Ergonomics. 36, 9, 1099–1110 (1993).

[3] Brookings, J., Wilson G., Swain C.: Psychophysiological responses to changes in workload during simulated air traffic control. Biol. Psychol. 42, 3, 361–377 (1996)..

[4] Buzan, T., Buzan, B.: The Mind Map Book. BBC Books, London (1993).

[5] Caire, P., Genon, N., Heymans, P., Moody, D.: Visual notation design 2.0: Towards user comprehensible requirements engineering notations. RE. pp. 115–124 (2013).

[6] Cockburn, A., Savage, J., Wallace, A.: Tuning and testing scrolling interfaces that automatically zoom. SIGCHI Conf. Hum. factors Comput. Syst. - CHI '05. (2005).

[7] Cook, C., Irwin, W., Churcher, N.: A user evaluation of synchronous collaborative software engineering tools. 12th Asia-Pacific Softw. Eng. Conf. (2005).

[8] Crosby, M.E., Stelovsky, J.: How do we read algorithms? A case study. Computer (Long. Beach. Calif). 23, 25–35 (1990).

[9] Diepenbeck, M. et al.: Behavior Driven Development for circuit design and verification. High Level Design Validation and Test Workshop (HLDVT), 2012 IEEE International. pp. 9–16 (2012).

[10] Emrich, M., Press, D.: Behaviour Driven Development with JavaScript: An Introduction to BDD with Jasmine. Developer Press.

[11] Hart, S.G.: Nasa-Task Load Index (NASA-TLX); 20 Years Later, (2006).

[12] Hart, S.G., Staveland, L.E.: Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. In: Hancock, P.A. and Meshkati, N. (eds.) Human Mental Workload. pp. 139–183 North-Holland (1988).

[13] Morrison, P., Holmgreen, C., Massey, A., Williams, L.: Proposing regulatory-driven automated test suites for electronic health record systems. Software Engineering in Health Care (SEHC), 2013 5th International Workshop on. pp. 46–49 (2013).

[14] NEČAS, I.: BDD as a Specification and QA Instrument [online]. Masarykova univerzita, Fakulta informatiky.

[15] North, D.: Introducing BDD, http://dannorth.net/introducing-bdd/, (2013).

[16] Oakley, I., Brewster, S., Gray, P.: Can you feel the force? An investigation of haptic collaboration in shared editors. proceedings of EuroHaptics. (2001).

[17] Paas, F., Tuovinen, J., Tabbers, H., Gerven, P.: Cognitive load measurement as a means to advance cognitive load theory. Educ. Psychol. 38, 1, 63–71 (2003).

[18] Rastkar, S., Murphy, G., Bradley, A.: Generating natural language summaries for crosscutting source code concerns. IEEE International Conference on Software Maintenance, ICSM. pp. 103–112 (2011).

[19] SDMetrics: SDMetrics, http://www.sdmetrics.com, (2014).

[20] Solis, C., Wang, X.: A Study of the Characteristics of Behaviour Driven Development. 37th EUROMICRO Conference on Software Engineering and Advanced Applications. IEEE Computer Society, Washington, DC, USA (2011), pp. 383–387.

[21] Wanderley, F., Silva, A., Araújo, J, Silveira, D.: SnapMind: A framework to support consistency and validation of model-based requirements in agile development. MoDRE 2014, Karlskrona, Sweden. pp. 47–56 IEEE (2014).

[22] Wilson, G.F.: An Analysis of Mental Workload in Pilots During Flight Using Multiple Psychophysiological Measures. Int. J. Aviat. Psychol. 12, 1, 3–18 (2002).

[23] Kruskal, W., Wallis, A.: "Use of Ranks in One-Criterion Variance Analysis". Journal of the American Statistical Association. 47.260 (1952), pp. 583–621.

[24] Mann H.B., e D. R. Whitney, D. R.: “On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other”. The Annals of Mathematical Statistics 18.1 (1947), pp. 50–60.

[25] I. Mahmud “Mind-mapping: An Effective Technique to Facilitate Requirements Engineering in Agile Software Development” (ICCIT) 2011.

[26] L. Kof., R. Gacitua, M. Rouncefield, P. & Sawyer. “Concept mapping as a means of requirements tracing” (MARK) 2010.

[27] P. Laplante. “What Every Engineer Should Know about Soft. Engineering”. 2007

[28] J. Village, F. Salustri, P. Neumann. “Cognitive mapping: Revealing the links between human factors and strategic goals in organizations” Journal Ergonomics. 2013.

[29] I. Alexander and N. Maiden, “Scenarios, Stories and Use Cases: Through the Systems Development Life Cycle”, John Wiley & Sons - ISBN 0-470-86194-0, 2004.

[30] A. Sutcliffe, S. Thew, P. Jarvis. “Experience with user-centred requirements engineering”. Requirements Engineering Journal vol. 16 - Springer Editions. 2011.

[31] Nielsen, J. Usability Engineering. Academic Press. 1993.