Building a Unified Ontology for Behavior Driven Development Scenarios

Konstantinos Tsilionis, Yves Wautelet(✉), and Samedi Heng

KU Leuven, Leuven, Belgium
{konstantinos.tsilionis,yves.wautelet,samedi.heng}@kuleuven.be

Abstract. Behavior Driven Development (BDD) offers a way to write scenarios in structured natural language on how to successfully fulfill a requirement. We fail to find documentation on how to use existing BDD templates. A set of templates with a clear definition of the keywords to use would provide guidance. This paper empirically explores the keywords found in the different dimensions of BDD scenarios to build a reference set of non-redundant concepts.

1 Introduction and Research Approach

A Behavior Driven Development (BDD) scenario describes a way to *execute the requirement* depicted in a user story. Tsilionis et al. [7] presents the first version of an ontology depicting the keywords most usually found in BDD templates without details on how it was built. This paper describes these; to this end, we applied a method similar as in Wautelet et al. [8] consisting of collecting and associating semantics to the most frequently found keywords in the *GIVEN*, *WHEN* and *THEN* dimensions.

1.1 *Descriptive_Concepts* in BDD Test Scenarios

To build the ontology, the goal is to collect the keywords and thus the concepts that are effectively used in practice when building BDD scenarios and to bring more formality and consistency in their use. The research process first required to collect data; the latter was gathered online in order to list and evaluate the most commonly used BDD test scenario templates. Scenarios are typically structured around the *GIVEN*, *WHEN*, and *THEN* dimensions (these will be referred to as the BDD scenarios' *dimensions* in this study). We consider each keyword found in such BDD templates as a *Descriptive_Concept* (*D_C*) which is a class of concepts containing (as attributes) a dimension (GIVEN, WHEN or THEN), a syntax (i.e. the keyword itself) and a semantic (a definition). The *D_C*-based approach was defined and applied in Wautelet et al. [8]. *D_C* as well as their dimension and syntax attributes can immediately be instantiated when a template is found in a formal or informal source (so typically we have one instance per dimension). Further investigation is generally needed to fill out the *semantic* attribute. We seldom find a definition associated to a keyword so it needs to be associated with it in another way (this is documented in Sect. 2).

D. Taibi et al. (Eds.): PROFES 2022, LNCS 13709, pp. 518–524, 2022.
https://doi.org/10.1007/978-3-031-21388-5_36

1.2 Building the Dataset

This section depicts the process of collecting data to gather the most commonly used **test scenario templates** used by BDD practitioners. We distinguish between *formal sources* (i.e., published scientific articles and books on BDD, see Appendix A[1]) and *informal sources* (i.e., blogs and forums addressing BDD, see Appendix B). Overall, these primary data sources yielded 120 formal and informal BDD test scenario templates widely used (see Appendix E). The formal sources came from searches on Google Scholar, Limo libis, IEEE Xplore and Springer Link using the keywords "scenario acceptance test", "bdd", "gherkin", "given when then", "behavior driven development", "bdd scenario". The first 10 pages of the returned results, per source, were consulted. The templates extracted from these sources can be found in Appendix A. Informal sources were found them using the same keywords as for formal sources but also including the following ones: "feature file", "bdd feature file", "feature file template", "bdd template", and "scenario template"; we used the Google search engine. As for the formal sources, the first 10 pages of the returned results were consulted. The templates extracted from these sources can be found in Appendix B.

During the elaboration of the dataset, each element that we find in a BDD scenario template that relates to one of the three dimensions becomes an instance of the *D_C* class. As an example, for the template 'GIVEN <*a context*>, WHEN <*an event*>, THEN <*an outcome*>', we will have three instances: one for *context*, one for *event*, and one for *outcome*. Each of these instances is related to the corresponding dimension of the template; the attribute *dimension* of the *D_C* class must therefore take one of the values *GIVEN*, *WHEN* or *THEN*. The attribute *syntax* will take the term found within the dimensions themselves (i.e., **context** for GIVEN, **event** for WHEN and **outcome** for THEN). Finally, the attribute *semantic* will be instantiated later through the use of publications addressing agile processes, Goal-Oriented Requirements Engineering (GORE) frameworks and other references in requirements/software engineering.

1.3 Building the Ontology

The elaboration of our data sources (formal and informal) yielded 120 test scenario templates containing multiple keywords. Each keyword has been considered separately and included in a list related to the dimension it supports. From that point, a series of refinements were made to keep the most relevant keywords. Relevant means here precise, specific and complementary to the other keywords ensuring the coherence of all the scenarios' dimensions. More specifically, these refinements were necessary to i) filter-out non-significant/vague/overlapping keywords allowing the remaining ones to serve as the candidate *D_C* for inclusion in an ontology, and ii) associating a semantic to each of the candidate *D_C*. The refinement process is described below.

First, on the basis of the dataset, we listed all of the keywords in a table where each dimension is considered separately. The number of occurrences of the keyword in formal and informal sources was noted; In total, 21 different instances were

[1] Appendices are consolidated in Appendix_Consolidated_BBD_templates.docx. at: https://data.mendeley.com/datasets/svmcxt5z5f/1.

recorded for the *GIVEN* dimension, 22 for the *WHEN* and 19 for the *THEN* dimension (see Appendix F). Next, informal non-significant and vague terms were removed (e.g. 'Something', 'Scenario', 'It', 'Future', 'Past', 'Present', etc.). We then associated semantics to all of the potential *D_C* instances. Since no semantics were ever found with the collected templates, we had to find semantics in another way. A first overview has been done in BDD related books to evaluate if more information on templates was available. We looked for definitions of the keywords, found in the previous stage, in a list of sources in the domain of agile processes, GORE frameworks, and software engineering to find a matching semantic. When a match was found between the syntax appearing in a test scenario template dimension and a semantic given in the former sources, we proceeded to a preliminary adoption and did not go through the rest of the sources in the list. The keywords for which we could associate a semantic were allowed to proceed to the next stage as *D_C* candidates. Otherwise, the keyword was being abandoned and considered irrelevant. The list of sources from the most to the least preferred one were: (i) User Stories Applied: a publication elucidating the ways for improvements in agile processes in requirements engineering [1]; (ii) KAOS: a framework for requirements engineering based on goal modeling [2,4]; (iii) Requirements Engineering Fundamentals: a study guide for the Certified Professional for Requirements Engineering Foundation Level exam as defined by the International Requirements Engineering Board (IREB) [5]; (iv) BABOK: a professional guide describing the terms and concepts related to the role of a business analyst [3]; and (v) SEVOCAB: a glossary of concepts and their definition in the field of Software and Systems Engineering [6]. Next, we compared the semantics associated to the keywords that were retained in the previous stage. This was done to highlight any similarities/overlapping/mismatches between semantics into a same dimension. Explicitly, every initial semantic overlap between two (or more) keywords was further analyzed. In several occasions, a presumed semantic overlap was eventually being dismissed as one upon further investigation. Each *D_C* instance candidate was then allowed to pass to the next stage of evaluation. If the semantic overlap was persisting, we were checking whether the use of another source from the aforementioned list could attribute a different semantic definition to either of the two (or more) keywords. The *D_C* instance of which the semantic was the most alienated to the purpose of the scenario's dimension was taking a new semantic from another source. If no new semantic could be allocated to the keyword through another source, the most generic one was retained. The kept *D_C* were included to form our base ontology and their semantics were then evaluated one last time on the basis of the secondary data, i.e. the set of test scenario examples. Few *D_C* remained at the end of this process; they were consolidated as an ontology.

2 Building the Ontology for BDD Scenarios

A Table summarizing the relevant keywords from each BDD scenario template found in the primary data set can be found in Tsilionis et al. [7]. We discuss in this section the choices that have been made to select *D_C* instances for the 3 dimensions.

2.1 The *GIVEN* Dimension

Syntax Included and Semantic Association: Using the method and the list of sources depicted in Sect. 1.3, the semantics associated to the kept syntaxes were: (i) Context: *The system context is the part of the system environment that is relevant for the definition as well as the understanding of the requirement of a system to be developed* [5]; (ii) Precondition: *A required precondition captures a permission to perform the operation when the condition is true* [4]; (iii) State: *A state defines a period of time in which a system shows a particular behavior and waits for a particular event to occur* [5]; and (iv) Input: *An input represents the information and precondition necessary for a task to begin; it may be: explicitly generated outside the scope of business analysis (e.g. construction of a software application) or generated by a business analyst task* [3].

Comparison of Associated Semantic: A complementarity was noted between the semantics associated to the keywords *Precondition* and *Input*. More detailed, the International Institute of Business Analysis (IIBA) [3] states that an input can be regarded as a precondition to start a task; all in all the *Precondition* encompasses the *Input* but is more general than it so we decided to keep the former as one of the *D_C* candidates to be integrated in the ontology. Additionally, *State* and *Context* are both described in [5] as (system) behavior-communicating elements. However, the former seems to focus on the time-dimension of the system's expressed behavior in-between transitions while the latter focuses on the system's surrounding circumstances to better understand the behavior itself. Therefore, despite their slight initial convergence in their meaning, these two elements seem not to be overlapping each other. To be sure, we allowed the *D_C* class instantiated with both of these keywords to proceed to the next stage so they can be further evaluated semantically based on our assembled BDD scenario examples.

Semantic Evaluation on Examples: The semantics for *Context*, *Precondition*, and *State* were further evaluated on the basis of BDD scenario examples gathered from our secondary dataset. This revealed that the word *Precondition* was used in 59% of the scenarios' instances, compared to a corresponding 25% use of the word *Context* and 16% use of the word *State*. Despite the predominance of the word *Precondition* compared to the other two terms, their semantic interpretation could not be easily differentiated within the examples where it was suggested that *Context* was incorporating a set of necessary *Preconditions* required for the BDD testing phase landing the system in a specific *State*. The *State* is the examples related to a set of *Preconditions* rather than behavior as suggested in its definition. We decided thus to keep the *State* element but to change its semantics to "a set of preconditions" rather than the original semantics that were associated to it in order to match the empirical use of the term. Hence, all three concepts were kept as candidates for the ontology.

2.2 The *WHEN* Dimension

Syntax Included and Semantic Association: Using the method and the list of sources depicted in Sect. 1.3, the semantics associated to the kept keywords were: (i) Event: *Actions and events are the plot of a scenario. They are the steps an actor can take to achieve his goal or a system's response* [1]; (ii) Action: *Actions and events are the plot*

of a scenario. They are the steps an actor can take to achieve his goal or a system's response [1]; (iii) Interaction: *An interaction is an action that takes place with the participation of the environment of the object* [6]; (iv) Behavior: *Observable activity of a system, measurable in terms of quantifiable effects on the environment whether arising from internal or external stimulus* [6].

Comparison of Associated Semantic: Sevocab [6] details that an *Interaction* can be uni-directionally regarded as an *Action* while the opposite does not seem to hold. Hence, out of the two, the latter being more generic, it seems like the better candidate for a possible integration in the ontology. Moving on, Cohn [1] yields an exact overlap between the semantic definition of *Event* and *Action* so we had to proceed to the next source to see whether the meaning of the two could be extended further. The IIBA [3] describes an *Event* as a *system trigger initiated by humans* whereas Darimont et al. [2] describe an *Action* as *an input-output relation over objects; action applications define state transitions; actions may be caused, stopped by events and they are characterized by pre-, post- and trigger-conditions*. So Darimont et al. [2] present actions to be initiated by events rendering the latter as a trigger of the former; with their semantic being aligned it is equal to take one or the other but one must be selected. So the *Event* was allowed to move on to the next phase of evaluation as a candidate *D_C*.

Semantic Evaluation on Examples: The words *Event* and *Behavior* were prevalent in the test scenario examples (76% and 22%). Also, 2% of the examples contained the word *Precondition*, but the last one was not part of our primary syntax selection for this dimension so it was not further considered. Given the clear predominance in the use of the word *Event* within the examples, corresponding also to the semantic definition as prescribed in the previous phase, we decided to keep this syntax as candidate for the *D_C* instance for this dimension. The term *Behavior* was also kept because of the clear difference in its definition with respect to the other concepts.

2.3 The *THEN* Dimension

Syntax Included and Semantic Association: Using the method and the list of sources depicted in Sect. 1.3, the semantics associated to the kept syntaxes were: (i) Outcome: *The business benefits that will result from meeting the business needs and the end state desired by stakeholders* [3]; (ii) Postcondition: *A required postcondition captures an additional condition that must hold after any application of the operation* [4]; (iii) Output: *An output is a necessary result of the work described in the task. Outputs are created, transformed or change state as a result of the successful completion of a task* [3]; (iv) Change: No semantic was found so it was considered non-relevant.

Comparison of Associated Semantic: A semantic complementarity was noted between *Outcome* and *Output* as the IIBA [3] portrays both as the *culminating effect* of a task/operation. This similarity can be problematic as no clear differentiating factor can be found between these *D_C* instances so we proceeded to the next source seeking whether the meaning of the two can be extended. Sevocab [6] defines *Outcome* as *an artefact, a significant change of state or the meeting of specified constraints* and *Output* as a *a product, result or service generated by a process* or as an *input to a successor*

process. The latter definition outlines the process-driven nature of an *Output* signaling a temporary result being in a transient state while waiting to contribute as input to the start of the next activity; on the other hand, an *Outcome* is deemed as an enduring effect signifying the achievement of a specific purpose. Considering the culminating disposition of the *THEN* dimension in a BDD scenario, we considered the instance of the *D_C* class associated to the syntax *Outcome* as more relevant for constructing the ontology.

Semantic Evaluation on Examples: Our consulting examples depicted a 57% use of the term *Postcondition* compared to a 29% use of the word *Outcome*. They also showed a 14% use of the word *Event* but as the last one was not part of the selection process for the *THEN* dimension, it was not considered further. Despite the predominance of the term *Postcondition*, we encountered difficulties dissociating it from a *State* in the sense that one or multiple postconditions were required to be satisfied for the achievement of an outcome within the examples. Hence, both *D_C* instances through their associated semantics were considered relevant for the construction of the ontology.

3 Ontology for BDD Test Scenarios

The remaining concepts have been placed in an ontology. From the selection process, two kind of concepts can be distinguished: *user-driven* and *system-driven* scenarios. The former refer to human-related concepts, i.e., the *Context*, the *Event* and the *Outcome*. These are typically instantiated by depicting the behavior taken by the user to achieve the outcome; these are expressed using a pronoun. Conversely, the system-driven concepts refer to software related concepts, i.e. the *Precondition*, the *Behavior* and the *Postcondition*; these are typically instantiated by describing successively the state of the system before, and after the occurrence of a specific event. In the ontology, the keywords *Behavior* and *Event* are difficult to evaluate (and differentiate) in nature without their associated semantics. Moreover, the keyword *Behavior* is misleading since it refers to system behavior in the semantics but, by nature, it is matching to the topic of behavior driven development which is theoretically centered on the user. The true element that assists in the discrimination of instances is the *WHEN* dimension so that particular attention needs to be dedicated to it. We thus change the keyword *Event* to *User_Behavior* and the keyword *Behavior* to *System_Behavior* while keeping their associated semantics. Finally, a *State* is seen as a set of preconditions; this is here also extended to the postconditions. The *State* thus only concern the system-driven context.

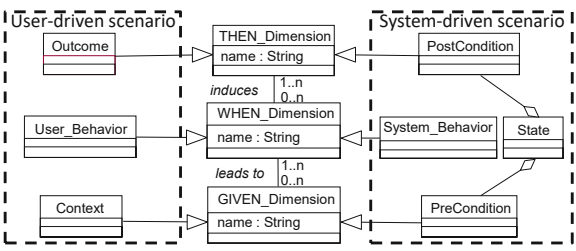

Fig. 1. Ontology for BDD test scenarios.

References

1. Cohn, M.: User Stories Applied: For Agile Software Development. Addison-Wesley (2004)
2. Darimont, R., Van Lamsweerde, A.: Formal refinement patterns for goal-driven requirements elaboration. ACM SIGSOFT Softw. Eng. Notes **21**(6), 179–190 (1996)
3. IIBA, K.B.: A Guide to the Bus. Anal Body of Knowledge. International Institute of Bus (2009)
4. Letier, E., Van Lamsweerde, A.: Deriving operational software specifications from system goals. ACM SIGSOFT Softw. Eng. Notes **27**(6), 119–128 (2002)
5. Pohl, K.: Requirements Engineering Fundamentals: A Study Guide for the Certified Professional for Requirements Engineering Exam-foundation Level-IREB Compliant. Rocky Nook, Inc. (2016)
6. SEVOCAB: Software and Systems Engineering Vocabulary. IEEE Computer Society (2015)
7. Tsilionis, K., Wautelet, Y., Faut, C., Heng, S.: Unifying behavior driven development templates. In: 29th IEEE International Requirements Engineering Conference, RE 2021, pp. 454–455. IEEE (2021)
8. Wautelet, Y., Heng, S., Kolp, M., Mirbel, I.: Unifying and extending user story models. In: Jarke, M., et al. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 211–225. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_15