

Generation of Multiple Conceptual Models from User Stories in Agile

Abhimanyu Gupta
Ghent University
Gent, Belgium
Abhimanyu.Gupta@ugent.be

Abstract

While agile methodologies are commonly used in software development, researchers have identified many issues related to the requirements elicitation in agile projects. Some of these issues relate to documentation and more specifically the development, maintenance, and management of user stories. This research addresses some of the user stories challenges by proposing use of conceptual models while development of user stories. Conceptual models are visual representations that are commonly used for understanding the domain of business functions and communicating with the stakeholders. The research considers development of such conceptual models automatically (with the help of a tool) while user stories are developed. Such conceptual models can provide rich perspectives of the domain from multiple views (e.g. structural and dynamic). A detailed research plan has been developed to conduct this research.

Keywords: user stories, agile methodologies, conceptual models

1 Introduction

In Agile software development processes, the software requirements documentation is limited to the creation of user stories [1]. The user stories are simple description of a feature of the working software written from the user's perspective [2, 3].

Because of the substantial number of user stories that are developed in an Agile software development project, the Agile team finds difficulty in maintaining, tracing, and managing the user stories [4]. Even for moderately complex software, the number of user stories easily exceeds human capacity of overview and understanding. To alleviate this problem, we suggest using a tool to automatically generate conceptual models in the process of developing user stories. Conceptual models are visual representations that are commonly used for understanding the domain of business functions and communicating with the stakeholders [5].

Copyright © by the paper's authors. Copying permitted for private and academic purposes.

In: P. Mäder, P. Spoletini (eds.): Joint Proceedings of REFSQ-2019 Workshops, Doctoral Symposium, Live Studies Track, and Poster Track, Essen, Germany, 18-03-2019, published at <http://ceur-ws.org>

1.1 Research Problem

There has been limited research in developing conceptual models automatically from user stories. For example, Mesquita et al. [6] automatically extracted goal models (in i* language) from user stories. Robeir et al. [7] built a tool that automatically generates an ontological model of the domain (in OWL ontology) from user stories. And Lucassen et al. ([8]) developed a visual narrator tool to visually show concepts and relationships extracted from user stories. However, each of these tools have targeted only one type of conceptual model.

What is currently missing in the state-of-the-art is the automated generation of conceptual models that show both the functional (de)composition of the domain (i.e. the process view) and the domain concepts and the relationships (i.e. the object view). Moreover, the tools have focused on the standard structure of user stories structure which is: As a <type of user>, I want <some goal> so that <some reason>.

In this paper, we extend the current research by extracting multiple conceptual models from user stories. Multiple conceptual models provide rich perspectives of the domain from multiple views (e.g. structural perspective and dynamic perspective of a domain) [9]. We achieve this by adding acceptance criteria to the standard template of user stories which allows us extraction of multiple conceptual models from the user stories. Acceptance criteria are the performance or other metrics that define the acceptable functionality of a story [3]. The conceptual models that are developed by our tool are: ER diagram, BPMN process model, UML State Machine model, and UML Use Case model.

1.2 Relevance

While use of conceptual models is popular in the industry, there is also a growing evidence of use of multiple conceptual models simultaneously. Surveys conducted on the use of multiple conceptual models indicate that practitioners indeed use more than one conceptual model for different types of tasks [13-15]. Recker [10] found that process modelers access additional grammars when modeling business processes. Similarly, Green et al. [16] found that a significant percentage of analysts use multiple conceptual models when performing requirements analysis tasks. Dobing and Parsons [14] mentioned that 90% of UML users employ at least two different UML grammars in at least one-third of their projects.

The reason why practitioners use multiple interrelated conceptual models is because information systems are getting more complex and interrelated models can be used to represent different aspects of the system [9]. Empirical studies on multiple conceptual models [11, 12] suggest using multiple models results in better performance than using a single model. Kim et al. [11] mention that complex IS should be represented using multiple interrelated conceptual models.

Research also investigated on why analysts use multiple conceptual models. Jabbari et al. [9] mention that multiple models are used to obtain multiple perspectives of the systems. Recker et al. [10] suggest using multiple conceptual models as no one model is a complete representation of a domain as no single available grammar is ontologically complete. Most grammars have been developed keeping in mind modeling a real-world phenomenon.

2 Research Plan

The research methodology will be implemented using the following stages:

Stage 1. In agile development, where there is less focus on documentation, it will be unrealistic to expect that users will develop and maintain the conceptual models in the process of creating the user stories. Therefore, a new prototype tool will be developed that will automatically create and update conceptual models when user stories are fed to it. The objective of the prototype will be to demonstrate the feasibility of creation of such tool. Once the prototype tool is developed, a validation of the tool in terms of feasibility and accuracy will be performed using a case study.

Stage 2. To test the recommendation that conceptual models can be helpful for developing user stories, a laboratory study will be done with real users as subjects. Subjects will be asked to extend and develop a set of user stories from a specific set of existing user stories. A Factorial Design of Experiment study will be used where a group of subjects will have access to the conceptual models (related to the domain) and another group of subjects will not have any access to these models. An eye tracking device will be used to identify: (1) whether the conceptual models that are provided to one set of users are indeed being referred to develop the user stories, (2) if these models are used then which specific parts of the models are referred to, (3) between the two groups, whether there is a difference in the pattern of using the existing user stories.

2.1 Proposed Solution

We include the acceptance criteria based on Behavior-Driven Development (BDD), which is a set of software engineering practices designed to develop high quality software faster [17], in the standard user story template, as used in practice. It is based on agile practices followed in test driven development and domain driven design and provides a common language based on simple structured English that facilitates communication between project team members and business stakeholders [18]. The BDD scenario consists of a feature title, an associated user story and scenarios where each scenario is defined by three keywords – Given, When, and Then [3]. Given describes context, when specifies actions or events and then states expected outcomes of the performed actions. Following (Figure 2) is the BDD scenario template recommended by the agile community [18].

Table 1. The Behavior Driven Development (BDD) scenario template [18]

```
Feature: [title]
As [role] I want [feature] So that [benefit]
Scenario: [title]
  Given [context] And [some more context]
  When [some event occurs] And [some other event]
  Then [outcome] And [some other outcome]
```

As user stories are written in the context of users performing a certain task, therefore we base our construct in terms of agents. We develop an agent-based framework in three steps. First, we identify a set of basic constructs from an agent's perspective. Second, we identify the relationships that exist among these constructs. Third, we support the concepts and their relationships from the literature. These concepts and their relationships will provide theoretical framework for the algorithm.

To provide an overall view of the concepts and their relationships we depict the concepts and their relationships in a visual model represented in Entity-Relationship Diagram. We use the ER notation because it is widely familiar, simple, and often used in information systems analysis and design.

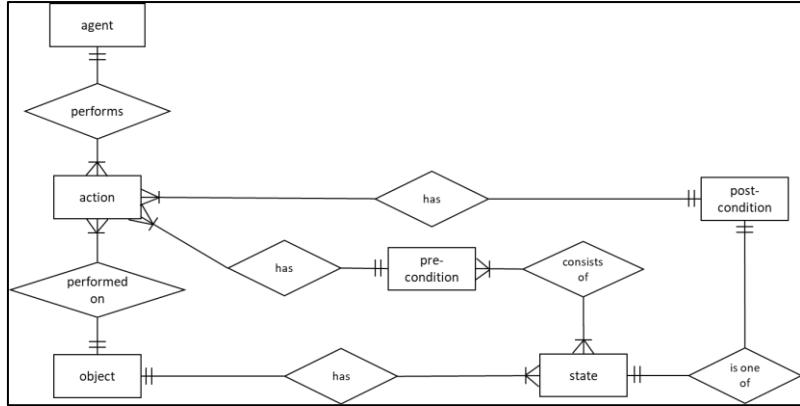


Fig. 1. ER Diagram describing entities and relationships of the agent-based framework

2.2 Novelty of the Solution

Our approach here is to not involve the users to focus on the models that are created rather focus on writing and maintaining the user stories in certain way so that the models are output of the user stories. In fact, users are not even aware what types of models can be created using the user stories. Our approach is based on the premise that users (e.g. business analysts) do not have additional time to learn and create conceptual models and they should rather focus on creating and maintaining the user stories properly.

We would like to develop interrelated models that complement each other in terms of representing different aspects of the domain. Booch, Rumbaugh, and Jacobson [19] provide a basic classification of conceptual models- behavioral and structural models. These two types reflect the dynamic and static nature of the systems respectively. From the user stories, we will develop conceptual models that are both behavioral and structural models. Table 2 shows the list of these conceptual models.

Table 2. Types of conceptual models developed from the tool

Conceptual Model	Type
Entity-Relationship (ER) Diagram(s)	Structural
Business process Modelling Notation (BPMN) Diagram	Behavioral
UML Finite State Machine (FSM) Diagram(s)	Behavioral
UML Use Case Diagram	Structural

2.3 Research Method

The objective of the research tool is to convert the knowledge contained in a set of user stories and corresponding acceptance criteria to various conceptual models. An Agent-Action framework is applied on the user stories to identify the agents, actions, objects. Then the relationships, dependencies and transitions are identified from the information available in the acceptance criteria. Finally, the four (4) types of diagrams are drawn.

Following are the steps involved in creating the conceptual models. The procedure starts by defining the valid indicators and splitting the user stories and their acceptance criteria in six (6) segments – Who, What, Why, Precondition, Action and Postcondition. After this step, the parts of speeches of all the words appearing in those six segments are identified. Then the identification of various components like agent, object, action, concept(s) of precondition, state of precondition, concept of postcondition and state of postcondition are completed and all information are stored into an intermediate table.

A thorough reconciliation takes place to ensure that all components are present in all user stories and their acceptance criteria. If components are not present, then they are populated from appropriate components of the previous user story. After this, the intermediate table is generated as one of the outputs of this algorithm. Then, using the intermediate table, the Entity-Relationship Diagrams, BPMN Diagram, Finite State Machine Diagrams and Use Case Diagrams are drawn. Note, there can be more than one FSM Diagrams generated depending on the nature of the user stories.

Eye tracking offers a window into how individuals read and scan information that is displayed to them [20]. Eye movements provide a valid measure of distribution of attention. By relating eye movements with tasks, one can obtain a picture of the decision-making process. A common eye movement metric is eye fixation that is often measured with respect to time or count [21]. Eye tracking metrics such as fixation duration can be used to identify which parts of the conceptual models' users are referring to while developing the user stories. Eye tracking devices also allow counting how many times the models have been accessed to and in what sequence. Fixation duration on overall conceptual models and specific parts of the model will provide insights on how the conceptual models have been used. Comparative fixation duration analysis between the two groups on existing user stories will throw light on the pattern of user stories usage.

2.4 Progress

A prototype tool has been developed that automatically creates and updates conceptual models. Python text analytics (NLTK) has been used to create the tool that takes user stories with acceptance criteria as inputs and the intermediate table and the four (4) conceptual models as output. Table 3 below represents a sample set of user story and Fig 2 and Fig 3 represents the corresponding conceptual models as outputs of the proposed algorithm.

Currently a case study is being performed to validate feasibility and accuracy of the proposed algorithm of creating multiple conceptual models. After the validation phase, a laboratory study of the conceptual models using eye tracking is planned in 2019. As part of the laboratory study, a set of user stories will be first shared with the subjects of the study and then they will be asked to write user stories (and acceptance criteria) for a set of new functions. An objective set of scoring rules will be developed and used to assess the quality of user stories written by the subjects and then a Factorial Design of Experiment will be performed where different participants will be aided with different combinations of conceptual models. An eye tracking metrics such as fixation duration will be used to determine how conceptual models are used by the subjects. Lastly, A statistical analysis will be performed on the collected data to determine which conceptual models have significant contributions in shared understanding.

Table 3. Sample User Stories

- | |
|--|
| <ol style="list-style-type: none"> 1. As a customer, I want to create a service request so that I can have my problem solved. Given that the customer is active, when he submits a service request then the service request should be submitted. 2. As a support assistant, I want to accept so that I can start working on it. Given it is submitted, when the team starts working on it then it is open. 3. As a support assistant, I need to resolve so that the customer can close the ticket. Given a service request is open, when the team resolves it, then it is fixed. 4. As a customer, I need to approve the service request so that it can be closed. Given a service request is fixed, when I approve it, then the service request becomes closed. |
|--|

5. As a customer, I need to reject the service request so that it can be re-opened. Given a service request is fixed, when I reject it, then the service request is open.
6. As a customer, I want to cancel a service request so that the team can focus on other active requests. Given a service request is submitted or closed when customer cancels it then it will be canceled.

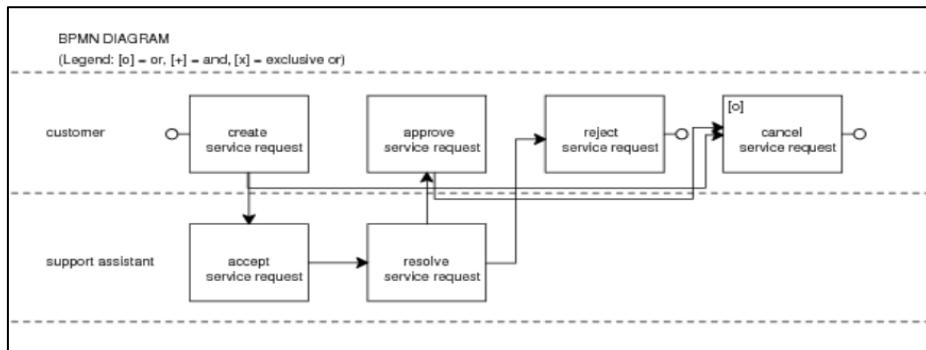


Fig. 2. Sample BPMN Diagram from User stories

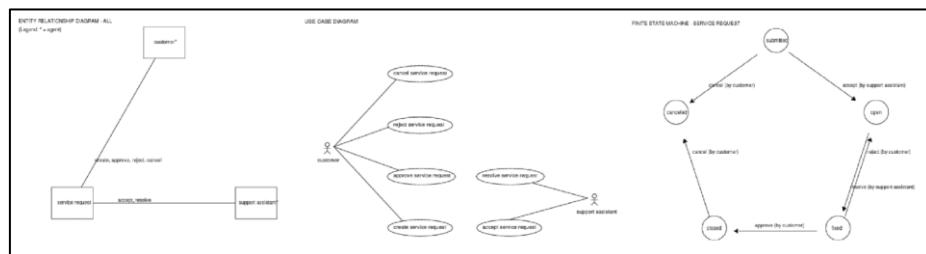


Fig. 3. Sample ER, Use Case and FSM Diagrams from User stories

References

1. Inayat, I., et al., *Review: A systematic literature review on agile requirements engineering practices and challenges*. Computers in Human Behavior, 2015. **51**(Part B): p. 915-929.
2. Leffingwell, D., *Agile Software Requirements: lean requirements practices for teams, programs, and the enterprise*. Agile Software Development Series. 2011, Boston: Addison-Wesley.
3. Cohn, M., *User Stories Applied: For Agile Software Development*. 2004, Boston: Addison-Wesley.
4. Ramesh, B., C. Lan, and R. Baskerville, *Agile requirements engineering practices and challenges: an empirical study*. Inf. Systems Journal, 2010. **20**(5): p. 449-480.
5. Wand, Y. and R. Weber, *Information Systems and Conceptual Modeling: A Research Agenda*. Information Systems Research, 2002. **13**(4): p. 363-376.
6. Mesquita, R., et al. *US2StarTool: generation i* models from user stories*. in *International i* Workshop (iStar)*. 2015.
7. Robeर, M., et al. *Automated extraction of conceptual models from user stories via NLP*. in *24th IEEE International Requirements Engineering Conference*, 2016

8. Lucassen, G., et al. *Visualizing User Story Reqs at Multiple Granularity Levels via Semantic Relatedness*. International Conference on Conceptual Modeling 2016.
9. Jabbari, S., A. Mohammad, and J. Recker, *Combined use of conceptual models in practice: An exploratory study*. Journal of Database Mgt., 2017. **28**(2): p. 56-88.
10. Recker, J., et al., *The Ontological Deficiencies of Process Modeling in Practice*. European Journal of Information Systems, 2010. **19**(5): p. 201-525.
11. Kim, J., J. Hahn, and H. Hahn, *How do we understand a system with (so) many diagrams? Cognitive integration processes in diagrammatic reasoning*. Information Systems Research, 2000. **11**(3): p. 284-303.
12. Siau, K. and L. Lee, *Are Use Case and Class Diagrams Complementary in Requirements Analysis? An Experimental Study on Use Case and Class Diagrams in UML*. Requirements Engineering, 2004. **9**(4): p. 229-237.
13. Davies, I., et al., *How do practitioners use conceptual modeling in practice?* Data and Knowledge Engineering, 2006. **58**(3): p. 358-380
14. Dobing, B. & J. Parsons, *How UML is used*. Comm. ACM, 2006. **49**(5): p. 109-113.
15. Fettke, P., *How Conceptual Modeling Is Used*. CAIS, 2009. **25**(1): p. 571-592.
16. Green, P. and M. Rosemann, *Integrated Process Modeling: An ontological evaluation*. Information Systems, 2000. **25**(2): p. 73-87.
17. Matula, J. and F. Hunka, *Enterprise Ontology-Driven Development*, in *Enterprise and Organizational Modeling and Simulation, Lecture Notes in Business Information Processing*, Pergl. R., et al., Editors. 2018, Springer.
18. Smart, J.F., *BDD in action: Behavior-Driven development for the whole software lifecycle*. Vol. 3. 2014, New York: Manning Publications Company.
19. Booch, G., J. Rumbaugh, and I. Jacobson, *The Unified Modeling Language User Guide*. 2 ed. 2005, Upper Saddle River, New Jersey: Addison-Wesley.
20. Rayner, K., *Eye movements in reading and information processing: 20 years of research*. Psychological Bulletin, 1998. **124**(3): p. 372-422.
21. Sharif, B. & J. Maletic. *An eye tracking study on the effects of layout in understanding the role of design patterns*. in *IEEE Int. Conf. on SW Maint.* 2010.