A Preliminary Framework for Constructing iStar **Models from User Stories**

Chunhui Wang¹, Chao Wu¹, Tong Li² and Zhiguo Liu¹

¹College of Computer Science and Technology, Inner Mongolia Normal University, 81 Zhaowuda Road, Hohhot 010022, China

²Faculty of Information Technology, Beijing University of Technology, 100 Ping Le Yuan, Beijing 100124, China

Abstract

User stories have been increasingly adopted in practice due to their intuitive structure, which specify users' needs for the software system from their perspectives. As iStar modeling framework also emphasizes user perspectives, some of its concepts and relationships can potentially be aligned with user stories. In this paper, we propose to (semi-) automatically derive iStar models based on user stories, with the aim of facilitating iStar modeling and promoting the practical adoption of iStar models. Specifically, this paper focuses on investigating an appropriate way of balancing the manual and automatic analysis during the modeling process, based on which we present a visionary framework.

Keywords

iStar, User story, Model construction

1. Introduction

User story is a widely adopted requirements notation in agile development. Generally, user stories are written by customers or users in natural language with limited format. For example, Cohn suggests a user story pattern[1]: As a <type of user>, I want <some intention>, so that *<some reason>*. Although a user story is short and simple, it describes a feature told from the perspective of the users/customers who desires the capability of the software system.

The iStar modeling technique is one of the most relevant goal-oriented requirements engineering approaches and provides a view of involved actors and their dependencies^[2]. iStar allows for the clear and explicit dependencies of goals to facilitate understanding of stakeholder needs, dependencies, etc. Through investigating the patterns of user stories, we argue that some of iStar concepts and relationships can potentially be aligned. For example, actor concept can be corresponded to type of user field, task or goal concept can be found from some intention and some reason fields, and relationships can be identified from the semantics of hierarchy and interdependence between user stories.

However, since user stories are short, simple descriptions and are generally presented in a flat list, it is difficult to identify iStar concepts and relationships from user stories. Existing research works have used requirements modeling methods, such as Rationale Tree[3] and Goal

© 02121 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0). CEUR Workshop Proceedings (CEUR-WS.org)

Proceedings of the 14th International iStar Workshop, October 18-21, 2021, St. Johns (NL), Canada EMAIL: ciecwch@imnu.edu.cn (C. Wang); wuchao@mails.imnu.edu.cn (C. Wu); litong@bjut.edu.cn (T. Li); cieclzg@imnu.edu.cn (Z. Liu)

Net[4], to model goals. These works use a graphical representation of user story sets to identify the hierarchy of goal concepts and their interdependence. In addition, because user stories are based on nature language expressions, some studies have used heuristic rules to automated goal model extraction using natural language processing[5]. However, due to the ambiguity of natural language descriptions, the precision results were still not as expected[6]. To ensure obtain practical adoption models from user stories, first, user stories should follow good writing specifications, and secondly, the nodes and edges that are automatically identified from use stories need to be confirmed manually. A corresponding research objective is how to maximally reduce such manual efforts and produce practically useful results.

This paper presents a preliminary framework for semi-automatically constructing iStar models from user stories. It is a human-assisted process, in which the concepts and relationships are first automatically extracted from user stories and then reviewed by analysts to generate iStar models. Specifically, this paper focuses on investigating the mapping between user stories and iStar models and providing an appropriate way of balancing the manual and automatic analysis during the modeling process, based on which we present a visionary framework.

2. Related Work

2.1. User Story

Follow the pattern of user stories, a user story contains three dimensions: **Who, What** and **Why**. In particular, in a user story (which is described with the way: As a <type of user>, I want <some intention>, so that <some reason>), <type of user> reflects the actor **who** uses the system, <intention> usually reflects **what** function/non-function the actor provides or uses, or **what** task/goal wants to achieve, and <some reason> usually reflects **why** the actor has the <intention> and indicates a goal or quality to achieve after the intention is finished.

User stories are subject to a process of sorting and dropping early on the development project[7]. From a macro-level perspective, there is a hierarchical relationship between user stories, that is, a user story may be an epic story, and the achievement of this story requires the completion of some *fine-grained* stories. In addition, there may be a temporal relationship between two user stories at the same grain, i.e., the realization of one user story depends on the other story. Thus, user stories can be grouped for serving for project composition/decomposition.

2.2. NLP for generating model from user stories

Natural language processing (NLP) for generating software model from user stories has become popular with the increase of agile software development, such as generating goal models[5], conceptual models[8] or UML models[9] from user stories. The main approaches commonly used were preprocessing, part-of-speech tagging (POS tag), syntactic parse tree and semantic analysis[6]. Preprocessing was used for preparing the data and usually included tokenization, filtering, and stop-word removal. POS tag was used to identify verbs and nouns as elements in models. Syntactic parse tree was normally used to represent a lexical category of a sentence and know grammatical relationships in the sentence. Several methods were used to obtain and understand the semantic connections in user stories, such as cosine similarity function and

clustering for semantic similarity. Some methods were used to identify topics inside user stories for heuristic analysis using LDA, Word Vectors and word embeddings. In our work, we will plan using NLP to perform lexical, syntax and semantic analysis of user stories, and automatically identify concepts and relationships to facilitate iStar model generation.

3. Proposal

The proposed framework for constructing iStar models from user stories comprises four functional modules: quality improvement, node identification, edge identification and model generation. Figure 1 shows the process modules and data flows within the framework. The function of each data processing module is described as follows:

- 1. Quality Improvement: automatically checks for irregularities in user stories, and promotes stakeholders to provide high-quality user stories.
- 2. Node identification: automatically identifies the concepts of iStar model from user stories.
- 3. Edge identification: automatically identifies the relationships between the identified concepts from user stories.
- 4. Model generation: requirements engineers or agile teams review the identified concepts and relationships and interactively generate iStar models.



Figure 1: The process framework of constructing iStar models from user stories

The framework shown in Figure 1 includes two iterative interactive activities. The first is that stakeholders modify user stories iteratively to ensure writing *good user stories*[10] with good readability and understandability. The second is that requirements engineers refer to the identified concepts and relationships and construct iStar model iteratively to ensure the precision of the model results.

In this work, a iStar model consists of a set of nodes that represent iStar model concepts and a set of edges that represent relationships among nodes, expressing that an actor wants/needs/would like to achieve an intentional element. The intentional element can be a resource, task, goal and quality. The details of the above concepts and relationships see [11].

3.1. Quality improvement

Although the user story has simple structure, there are often poorly written in practice and exhibit inherent quality defects[12]. We have carried out research work on improving the quality of user stories. In [13], we proposed 11 quality criteria to evaluate the quality defects of incomplete, inconsistent and untestable in user stories, and an approach based on model driven and NLP is proposed for finding the user stories with quality defects. In this work, these techniques in [13] will be used to ensure that users can write good stories.

3.2. Node identification

Nodes of iStar model can be aligned with the fields of **who**, **what**, **why** in user stories. In this work, user story can be written with the following grammar:

 As a <actor>, I want/want to/need/can/would like <task>/<goal>/<resource>/<quality>, [so that <task>/<goal>/ <quality>]

To improve the understandability of the transition from user story to the iStar nodes, the description (actor/task/capability/goal/quality) shall be kept short and precise. We advice these descriptions use the following syntax [14]:

- Actor: [Adjective]+ Noun;
- Task: Verb+Object+[Complement];
- Resource/Quality: [Adjective]+Object;
- Goal: Object + Passive_Verb + [Complement];

The node identification process mainly includes three steps. The first step is to identify the fields of **who**, **what** and **why**. The second step is to analyze the syntactic components of each field by preprocessing to get a set of sentences. The third is to obtain the concepts based on the result of matching syntactic components of the sentence in user story and the intentional element. In the third step, POS tag and syntactic parse tree are used in order to know the composition of the sentence.

3.3. Edge identification

According to [11], in this work, the edges in iStar mainly includes three types: *actors association links, social dependencies,* and *intentional element links.* To identify these edges, we provide corresponding heuristic rules and detection methods.

3.3.1. Actors association links

Actor links are binary, linking a single actor to a single other actor. Two types of actor links exist: *is-a* and *part-of*. To identified these actor links, we plan to adopt the method of *actor modeling*[1], that is, extract the actors/roles in all user stories, and then aggregate these actors according to their similarities, and recognize the links with the help of humans. In this phase, we plan to use the N-gram and WordNet to calculate the similarity among actors.

3.3.2. Social dependencies

Social dependencies are one-way, linking a *dependerElmt* within the *depender actor* to the *dependum* (an intentional element), outside actor boundaries, to the *dependerElmt* within the *dependee actor*. To identify the social dependencies, we plan to use the following two heuristics.

Heuristic Rule I: For a set of user stories, an object-based similarity calculation method is used to identify the *temporal relationship* between user stories[13], that is, if two user stories discuss similar objects, there may be social dependence between them, and the temporal order implies the direction between them.

Heuristic Rule II: For the two user stories u_1 and u_2 that have a temporal relationship, POS tag and syntactic parse tree are used to identify the corresponding concepts, and then the social dependency relationship is fed back. Among them, one of the description format of u_1 and u_2 is as follows:

- u₁: As a <depender>, I want/want to/need/can/would like <dependerElmt>, [so that <task>/<goal>]
- u₂: As a <dependee>, I want/want to/need/can/would like <dependum>, so that <dependeeElmt>

In some real case, some elements (*dependerElmt* and *dependeeElmt*) in the social dependency relationship may be omitted, and heuristic rules will be formulated for different omissions in our follow-up work.

3.3.3. Intentional element links

There are four types of links between intentional elements[11]: *neededBy*, *refinement*, *contribute* and *qualification*. We define the following rules and methods to identify these links.

Heuristic Rule III: For a user story pattern, As an <actor>, I want <resource> so that <task>, there is a *neededBy* link between a task and a resource.

To identify the *neededBy* link, POS tag and syntactic parse tree are used to find the concepts of *resource* and *task* and above the user story pattern.

Heuristic Rule IV: When there are hierarchical relationships between a goal (task) and some sub-goals (sub-tasks), there are *refinement* links between the goal (task) and the sub-goals (sub-tasks).

To discover the hierarchical relationships between goals/tasks, we plan to use a classification method based on themes which are extracted from user stories. This process mainly includes

two key steps. One is to extract terms from a set of goals (or tasks). The other is to classify goals/tasks using a text mining method based on topic modeling.

Heuristic Rule V: For a user story pattern, As an <actor>, I want <an intentional element> so that <quality>, there is a *qualification* link between an intentional element and a quality.

To the qualification links, we further recognize the *contribution* link, i.e., manually analyze the relationships from a source intentional element to a target quality and marked their types of *make*, *help*, *hurt*, and *break*[11] according to their semantics.

3.4. Model generation

Model generation is a human-assisted process in which requirements engineers view and modify the concepts and relationships identified by automated methods. We plan to design and implement a human-computer interaction environment. The environment will support the following functions: user story editor, user story quality improvement, iStar model concept and relationship recommendation, recommended information manual editor, and model visualization, etc.

The model generation process based on above human-computer interaction environment consists of four steps. The first step is stakeholders write user stories in user story editor. At the second step, the *user story quality improvement* module checks for and reports irregular and incomplete writing in user stories. Then, stakeholders will modify their user stories according to the feedbacks. The third step is iStar components in user stories are automated identified by the *iStar model concept and relationship recommendation* module and a preliminary iStar model is built. At the last step, the requirements engineers will review and modify the preliminary iStar model using text or graphic editing functions.

4. Conclusions

This paper presented a preliminary framework for constructing iStar models from user stories. The proposed framework consists of four modules, namely, quality improvement, node identification, edge identification and model generation. Quality improvement is used to find irregular user stories and improve the effect of automatically identifying concepts and relationships of iStar models. The identification process considers the potential mappings between the user stories and the model semantics. The model generation process is a human-assisted process based on manual and the automated identification results, which reduces the workload of manual modeling and promotes the practical adoption of iStar models. As for our next step work, we plan to develop a prototype tool that implements our proposed framework. In addition, we aim to conduct case studies to verify the effectiveness of the proposed approach.

Acknowledgments

This work is supported by the Project of Beijing Municipal Education Commission (No.KM202110005025) and the Natural Science of Foundation of Inner Mongolia Province (No.2021MS06024).

References

- [1] M. Cohn, user story applied for agile software development, Addison-Wesley Professional (2004).
- [2] J. Horkoff, F. B. Aydemir, E. Cardoso, T. Li, A. Maté, E. Paja, M. Salnitri, L. Piras, J. Mylopoulos, P. Giorgini, Goal-oriented requirements engineering: an extended systematic mapping study, Requirements Engineering 24 (2019) 133–160.
- [3] Y. Wautelet, S. Heng, M. Kolp, I. Mirbel, S. Poelmans, Building a rationale diagram for evaluating user story sets, in: 2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS), 2016, pp. 1–12.
- [4] J. Lin, H. Yu, Z. Shen, C. Miao, Using goal net to model user stories in agile software development, in: 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2014, Las Vegas, NV, USA, June 30 - July 2, 2014, IEEE Computer Society, 2014, pp. 1–6.
- [5] T. Güne, F. B. Aydemir, Automated goal model extraction from user stories using nlp, in: IEEE Requirements Engineering Conference, 2020.
- [6] I. K. Raharjana, D. Siahaan, C. Fatichah, User stories and natural language processing: A systematic literature review, IEEE Access 9 (2021) 53811–53826.
- [7] J. Patton, W. P. Economy, F. Fowler, A. Cooper, A. M. Cagan, User story mapping : discover the whole story, build the right product, O'Reilly Media (2014).
- [8] M. Robeer, G. Lucassen, F. Dalpiaz, S. Brinkkemper, Automated extraction of conceptual models from user stories via nlp, in: Requirements Engineering Conference, 2016.
- [9] A. Meryem Elallaoui, B. Khalid Nafil, A. Raja Touahni, Automatic transformation of user stories into uml use case diagrams using nlp techniques, International Conference on Ambient Systems, Networks and Technologies (2018) 42–49.
- [10] J. Patton, Telling better user stories, The software testing and quality engineering magazine 11 (2009) 24–29.
- [11] F. Dalpiaz, X. Franch, J. Horkoff, istar 2.0 language guide, Computing Research Repository (2016) 1–15.
- [12] G. Lucassen, F. Dalpiaz, J. M. E. M. van der Werf, S. Brinkkemper, Improving agile requirements: the quality user story framework and tool, Requirements Engineering 21 (2016) 383–403.
- [13] C. Wang, Z. Jin, H. Zhao, M. Chui, An approach for improving the quality of user stories, Journal of Computer Research and Development 58 (2021) 731–748.
- [14] G. Grau, X. Franch, E. Mayol, C. P. Ayala, C. Cares, M. Haya, F. Navarrete, P. Botella, C. Quer, Risd: A methodology for building i-strategic dependency models, International Conference on Software Engineering and Knowledge Engineering (2005) 259–266.