Using a Goal-Driven Approach to Structure User Story Sets

UU/SIKS Symposium on Natural Language in Requirements Engineering

Yves Wautelet
KU Leuven
Background: User Stories as Artefacts for Requirements Representation in Agile Methods

- Agile methods in general and XP in particular use user stories (US) to collect user requirements.
- These are mostly written in an informal manner; templates appeared over the years in the form:

  As [the WHO], I want/want to/need/can/would like [the WHAT], so that [the WHY]

Examples of user stories:

- As a user, I can backup my entire hard drive.
- As a power user, I can specify files or folders to backup based on file size, date created and date modified.
- As a user, I can indicate folders not to backup so that my backup drive isn’t filled up with things I don’t need saved.
- …
Templates to Write User Stories

- Generic Structure: *As [WHO], I want [WHAT], so that [WHY]*
- User story templates proposed by Mike Cohn:
  
  I as a `<role>`, I want `<function>`, so that `<business value>`.
  
  As a creator, I want to upload a video, so that any users can view it.

  As a `<type of user>`, I want `<capability>`, so that `<business value>`.
  
  As a book buyer, I want to search for a book by ISBN, so that I can find the right book quickly.

  As a `<type of user>`, I want `<some goal>`, so that `<some reason>`.
  
  As a user, I can indicate folders not to backup, so that my backup drive isn't filled up with things I don't need saved.
Open Issue in Agile Methods: How to Handle, Manage, Structure, (Re)present, Group, ... User Story Sets

US Templates

• US templates can be found in literature or are proposed by practitioners (notably into blogs)
• US templates introduce *Descriptive Concepts* into these templates in an ad hoc manner without defining them
• We thus dispose of templates with syntaxes associated to Descriptive Concepts but no semantics!
• Nevertheless, plenty of examples are always provided with the proposed templates.
Research Method

1. Building the Dataset
   - Formal Sources
   - Informal Sources
   - Collect User Story Examples

2. Building the Candidate Model
   - Evaluate User Story Examples with Semantic of Descriptive_Concept Candidates
   - Compare Semantics of Descriptive_Concept Instances among each other one of the same Dimension
   - Select and Associate Semantic with Syntax of Descriptive_Concept Candidates
   - i* framework, KAOS, BPMN, RE Glossary

3. Validation
   - Candidate Unified Meta-Model for US Concepts
   - Evaluate Candidate Model on Case Studies
Descriptive Concepts

• Each instance of the following class is a descriptive concept candidate as a concept class for our future unified model

<table>
<thead>
<tr>
<th>Descriptive_Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension : ENUM{WHO,WHAT,WHY}</td>
</tr>
<tr>
<td>syntax : String</td>
</tr>
<tr>
<td>semantic : String</td>
</tr>
</tbody>
</table>
Descriptive Concepts: Collected Syntaxes

- (number of occurrences found in formal sources + number of occurrences found in informal sources).

<table>
<thead>
<tr>
<th>WHO</th>
<th>WHAT</th>
<th>WHY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role (13 + 31)</td>
<td>Goal (4 + 18)</td>
<td>Business Value (7 + 18)</td>
</tr>
<tr>
<td>Type of User (8 + 15)</td>
<td>Something (3 + 10)</td>
<td>Benefit (7 + 18)</td>
</tr>
<tr>
<td>User (0 + 10)</td>
<td>Action (4 + 7)</td>
<td>Reason (4 + 14)</td>
</tr>
<tr>
<td>Actor (0 + 6)</td>
<td>Feature (4 + 7)</td>
<td>Goal (3 + 6)</td>
</tr>
<tr>
<td>System Role (0 + 1)</td>
<td>Function (1 + 7)</td>
<td>Achievement (0 + 4)</td>
</tr>
<tr>
<td>Persona (0 + 1)</td>
<td>Desire (0 + 6)</td>
<td>Rationale (0 + 2)</td>
</tr>
<tr>
<td>“x” (0 + 1)</td>
<td><strong>Functionality (1 + 4)</strong></td>
<td>Desire (0 + 2)</td>
</tr>
<tr>
<td></td>
<td>Capability (3 + 1)</td>
<td>Outcome (0 + 1)</td>
</tr>
<tr>
<td></td>
<td>Task (1 + 2)</td>
<td>Result (0 + 1)</td>
</tr>
<tr>
<td></td>
<td>Activity (1 + 2)</td>
<td>“z” (0 + 1)</td>
</tr>
<tr>
<td></td>
<td>Outcome (0 + 2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Behaviour (0 + 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Description (0 + 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>What (0 + 1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>“y” (0 + 1)</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Instances for Descriptive Concept and Related Syntax
Descriptive Concepts: Collected Syntaxes

• Descriptive Concepts with an insignificant number of instances were left out
• “Irrelevant” Descriptive Concepts (e.g. something, y, …) were left out of the model
Descriptive Concepts: Collected Semantics

- For each descriptive concept, semantics were looked after
  1. In the i* modeling framework
  2. In the KAOS framework
  3. In the Business Process Modeling Notation Framework
  4. A glossary of requirements engineering terminology

- When a match was found respecting the priority, we proceeded to a preliminary adoption

- A first comparison was made between the semantics to evaluate overlaps/redundancy.

- Further evaluation was then done on the collected examples

- Non redundant relevant elements were included in the candidate model (see paper for full discussion of the elements)
Unified Model
Adopted Semantics

- A role is an abstract characterization of the behavior of a social actor within some specialized context or domain of endeavor

- A task species a particular way of attaining a goal

- A capability represents the ability of an actor to define, choose, and execute a plan for the fulfillment of a goal, given certain world conditions and in the presence of a specific event

- A hard-goal is a condition or state of affairs in the world that the stakeholders would like to achieve

- A soft-goal is a condition or state of affairs in the world that the actor would like to achieve. But unlike a hard-goal, there are no clear-cut criteria for whether the condition is achieved, and it is up to the developer to judge whether a particular state of affairs in fact achieves sufficiently the stated soft-goal
Preliminary Case Studies: US issued of ClubCar and CalCentral (1/2)

- ClubCar is a multi-channel application available as an Android application, SMS service and IVR system. Users of ClubCar are riders and/or drivers, they can register by SMS, voice or through the Android app. Roughly speaking the software allows drivers to propose rides and submit their details with dates, times, sources and destinations while riders can search for available rides. The project included a total of 28 US.

- CalCentral is an online system that delivers a unified and personalized experience to students, faculty and staff, facilitating the navigation of campus resources, delivering personal notifications from key campus systems, and supporting learning and the academic experience. US are used as requirement artifacts in the project; the list of 95 US.
Fig. 4. Elements Coverage in the Carpooling and CalCentral Case Studies
Creating Visual Models Based on User Story Sets: The Rationale Diagram
Placing the unified model in the US-based development
Towards a visual representation of User Stories

• US tagging with the unified model furnishes information on the nature and grain of the US elements.

• We would like to use this information to graphically represent User Stories based on the former Unified User Story Model, so that we can **visualize** and **analyze** User Stories *(inter)dependencies.*
Rationale Diagram: i*-based Graphical Notation
### Rationale Diagram: a modeling example

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Element</th>
<th>D_C Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to register to the service</td>
<td>Task</td>
</tr>
<tr>
<td>WHY</td>
<td>so that I can propose ride to go from A to B</td>
<td>Hard-goal</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to propose a ride from A to B with the price location and time of departure, and number of seats available</td>
<td>Task</td>
</tr>
<tr>
<td>WHY</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to log in to the platform</td>
<td>Capability</td>
</tr>
<tr>
<td>WHY</td>
<td>so that I can register to the service</td>
<td>Task</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to select the ride characteristics</td>
<td>Capability</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to confirm the proposal</td>
<td>Capability</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want the RIDER to be satisfied of my service</td>
<td>Soft-goal</td>
</tr>
</tbody>
</table>
Rationale Diagram: a Modeling Example
**Rationale Diagram: Identifying elements from EPIC US**

- **EPIC**: top-level *Task* element not issued of the refinement of another *Task* element but that itself needs to be refined in more elements (*WHAT Dimension*).

Top-Level Hard-goal, One Means-End Decomposition
Rationale Diagram: Identifying elements from EPIC User Stories
Integration of the Rationale Tree in the SCRUM Board and Propagation Algorithm Based on Business Value

- Increase traceability and visibility on requirement elements across iterations and monitor the progress on multiple levels (i.e. the levels of the elements in the tree).
Rationale Tree: Application on a real life case study in the field of travel and expenses management.
(Preliminary) Results of the Real-Life Case Study

• Allows reasoning and evaluating the consistency in requirements
• Allows iterative planning based on business value
• Evaluation of the business value of elements starts with the fine-grained elements (presented at the top)

A full study of the results, lessons learned and threats to validity should still be performed!
Supporting CASE-Tool
Using the Rationale Tree for Generating an Agent-Based Design: Process Fragment Approach
Creating Visual Models Based on User Story Sets: Generating a Use Case Diagram
User story modeling with Use-Case diagram

Mapping between user story elements and Use-Case diagram elements

Role -> Actor
Hard-goal -> Use-Case
Task -> Use-Case

Capability -> X

Soft-goal -> RUP/UML Business Goal
## User story modeling with Use-Case diagram

### User story set from ClubCar

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Element</th>
<th>D.C Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role, Task</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to register to the service</td>
<td></td>
</tr>
<tr>
<td>WHY</td>
<td>so that I can propose a ride to go from A to B</td>
<td>Hard-goal</td>
</tr>
<tr>
<td>WHO</td>
<td>I want to propose a ride from A to B with the price location and time of departure, and number of seats available</td>
<td>Role, Task</td>
</tr>
<tr>
<td>WHAT</td>
<td>As a DRIVER</td>
<td></td>
</tr>
<tr>
<td>WHY</td>
<td>I want to log in to the platform</td>
<td>Capability, Task</td>
</tr>
<tr>
<td>WHO</td>
<td>As a RIDER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to be transported from A to B</td>
<td>Hard-goal</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want to confirm the proposal</td>
<td>Capability</td>
</tr>
<tr>
<td>WHO</td>
<td>As a DRIVER</td>
<td>Role</td>
</tr>
<tr>
<td>WHAT</td>
<td>I want the RIDER to be satisfied of my service</td>
<td>Soft-goal</td>
</tr>
</tbody>
</table>

![Use-Case Diagram](image)
User story modeling with Use-Case diagram

User story model with Use-Case diagram: Supporting CASE-Tool
Conclusion and Future Work
Conclusion

• We have build a unified model for user story templates with a limited set of concepts with defined syntax and semantics

• A User Story Set tagged using the unified model can be used for visual requirements representation

• The technique has been integrated in scrum approach and applied on a real life case

• The visual model can be used for forward engineering
Future Work

• The requirements modeling approach has been applied on case studies
  o a study of the success (and failure) criteria for its application can/should be performed

• A comparison between the rationale tree approach and the Quality User Story (QUS) framework developed at Utrecht University
  o Which approach is more efficient to report/detect defects or errors within user stories using the QUS framework and the rationale diagram?
  o Which approach is the most efficient to identify missing requirements?
  o Which approach allows to at best identify which functional and design choices best support the fulfilment of non-functional requirements.
  o Which approach best structures user stories into themes?
References


