Rule-Based Extraction of Goal-Use Case Integrated Models from Text

Huan Nguyen, John Grundy and Mohamed Almorsy
Outline

Introduction
Motivating Examples
Approach
Current Results
Challenges
Future Work
Introduction

Goal and Use Case Modeling:
- System goals capture (Goal-directed RE)
- System-user interaction modeling (Use Cases)
- Why use?
  - Top-down and bottom-up traceability
  - Comprehensive view of system requirements
- How capture from natural language text – this paper
- How represent using integrated model – MODELS 2015
The Problem

- Goal-Use case identification and extraction from text is tedious, time-consuming and error-prone
  - Not all sentences in a document present goal and/or use case specifications
  - Multiple artifacts may be embedded in a single sentence
  - Relationships between artifacts are not explicitly specified
  - Extracted text is often not grammatically correct
  - Artifacts need to be classified
Motivating Examples (see paper!)

(S1) This software system will be a Social Networking System for travellers around the world. (S2) This system will be designed to improve the quality of travel planning by providing tools to assist in facilitating the communication between travellers. (S3) To improve the quality of travel planning, the system supports travellers to share and gain experiences while remaining easy to understand and use. (S4) More specifically, Travellers shall be capable of writing reviews and travel articles and participating in forum discussions.

Use case: Create Reviews

Brief Description: (S4) A traveller creates a review

Pre-condition: (S5) Before this use case can be initiated, the traveller has successfully logged into the system.

Initial Step-By-Step Description

Step 1. (i) The traveller selects to create a review.

Step 2. (i) System prompts the traveller to select a review category.

Step 3. (i) The traveller selects review category. (ii) The list of categories includes hotel, attraction, restaurant and tour.

Step 4. (i) System displays the suitable review creation form to the traveller.

Step 5. (i) The traveller enters the review content and submits it. (ii) A review content contains subject, overall rating, individual ratings and comment.

Step 6. (i) System validates the review. (ii) The validation should be completed within 1 second.

Step 7. (i) If the review content is valid, the system stores the review into the database and displays confirmation message to the traveller; (ii) else the system displays the errors to the traveller and the traveller repeats step 5.

Post-condition: (S6) After this use case is successfully finished, the new review is stored.
1. Identification of goal and/or use case from text

- Example 2: “This system will be designed to improve the quality of travel planning by providing tools to assist in facilitating the communication between travelers.”
1. Identification of goal and/or use case from text

- **Example 2:** “This system will be designed to improve the quality of travel planning by providing tools to assist in facilitating the communication between travellers.”

- Expected extracted goals:
  - Improve the quality of travel planning
  - Facilitate communication between travellers
1. Identification of goal and/or use case from text

- *Example 3*: “Step 7. *If* the review content is valid, the system stores the review into the database and display confirmation message to the traveller; *else* the system displays the errors to the traveller and the traveller repeats step 5.”
1. Identification of goal and/or use case from text

- Example 3 - *Expected extraction results:*
  
  **Step 7:** If the review content is valid, the system stores the review into the database
  
  **Step 8:** The system displays a confirmation message to the traveller

  **Extension:**

  **Condition:** The review content is not valid
  
  **Step 7.1:** The system displays errors to the traveller

  **Starting Step:** Step 7 – **Resuming Step:** Step 5
2. Identification of artifact relationships

- Example 4: “To improve the quality of travel planning, the system supports travellers to share and gain experiences.”
2. Identification of artifact relationships

- *Example 4:* To improve the quality of travel planning, the system supports travellers to share and gain experiences...

- **Expected extracted goals and relationships:**
  - *Parent goal:* Improve the quality of travel planning
  - *Sub-goal 1:* Travellers shall be able to share experiences
  - *Sub-goal 2:* Travellers shall be able to gain experiences
3. **Artifact Classification**

- *Example 5:* Improve the quality of travel planning – Business goal

- *Example 6:* Travellers shall be able to share experiences – Functional Feature Goal

- *Example 7:* The system shall be secure – Non-functional Product Goal
Motivating Examples (Cont’d)

4. Ensure textual artifacts are properly specified
   - *Example 8*: The traveller enters the review content and submits *it*
     - The traveller enters the review content
     - *The traveller* submits *the review content*
4. Ensure textual artifacts are properly specified

- *Example 9*: Travellers *shall be capable of* writing reviews

- Travellers *shall be able to write* reviews
Automated Support for the Extraction of Goal-Use Case Models from Text

- Takes textual requirements documents as input
- Identifies goals and use cases, and their relationships from the requirements document
- Classifies the extracted goals and use case components
- Ensures the extracted goal and use case specifications conform to our boilerplates (specification templates)
- Builds a goal-use case integrated model as output
Approach (Cont’d) – see paper for details

Requirements Document (.doc, .txt)

1. Pre-processing
2. Sections with Artefact Specs
3. Linguistic Analysis
4. Parse Trees & Dependency Trees
5. Artifact & Relationship Identification
6. Raw Artifact Specs & Relationships
7. Artifact Polishing

(Optional) Goal-Use Case Model Review

Extracted Goal-Use Case Model

Goal-Use Case Model Construction

Classified Artefacts

Artifact Classification

Polished Textual Specifications

Finalized Goal-Use Case Model

M indicates activities which are done manually
S indicates activities which are done semi-automatically

Unlabelled activities are carried out automatically by GUEST
Step1. Pre-processing

Format Requirements (caveats / assumptions)

- SRS in .txt or .doc
- No slash (“/”). i.e., “there is a link to author/reviewer information”
- Section numbers must be in strictly ascending order
- No “table of contents”
- Each use case is in its own section
- Each use case component has an indicator. i.e., “precondition”
- Important details in plain text (i.e., not in figures or tables)
Step 1. Pre-processing (Cont’d) – Automated Support

- Extract SRS to plain text (figures/tables removed)

- Extract text from each important section and associate it with possible artifact types

- Remove unnecessary and unprocessable details: ! ? … ()
Step 2. Linguistic Analysis

- Resolve coreferences
  - Use the Stanford Coreference Resolution System
  - The traveller enters the review content and submits it
    $\Rightarrow$ The traveller enters the review content and submits the review content

- Split sentences

- Generate a dependency parse tree for each sentence
Users shall be able to write reviews
Users shall be able to write reviews – tree in text format:

nsubj(able-4, Users-1)
aux(able-4, shall-2)
cop(able-4, be-3)
root(ROOT-0, able-4)
aux(write-6, to-5)
xcomp(able-4, write-6)
dobj(write-6, reviews-7)
Extraction Rule Example

\[ X = \{ \text{design I intend I target} \} \]
\[ Y / \text{VB} \]
\[ \text{root}(X) \]
\[ \text{nsubjpass}(X, \{ \text{system I project} \}) \]
\[ \text{auxpass}(X, \{ \text{be} \}) \]
\[ \text{xcomp}(X, Y) \]
\[ \rightarrow \text{root}(Y) ; \]
Extraction Rule Example – ‘Root’ Action Explanation

Original root is: Node 1. If an action assigns Node 2 as the new root, then:

- New root: Node 2
- Original root: Node 1
- All removed
- Node 2 as new root
- Node 1 as removed

Node 2

Node 3  Node 4  Node 5

Node 6

Node 7

Node 10

Node 11

Node 8

Node 9

...
Extraction Rule Application Example

Consider this sentence: The system will be designed to improve the quality of travel planning
Step 3. Artifact and Relationship Identification – Extraction Rules

The system will be designed to improve the quality of travel planning.

\[
\begin{align*}
\text{nsubjpass}(\text{designed-5, system-2}) \\
\text{auxpass}(\text{designed-5, be-4}) \\
\text{root}(\text{ROOT-0, designed-5}) \\
\text{xcomp}(\text{designed-5, improve-7}) \\
X = \{\text{design, intend, target}\} \\
Y/\text{VB} \\
\text{root}(X) \\
\text{nsubjpass}(X, \{\text{system, project}\}) \\
\text{auxpass}(X, \{\text{be}\}) \\
\text{xcomp}(X, Y) \\
\rightarrow \text{root}(Y); \\
X = \text{designed-5} \\
Y = \text{improve-7}
\end{align*}
\]
Outcome: “to improve the quality of travel planning” (“to” will be removed in the ‘polishing’ step)
Step 3. Artifact and Relationship Identification – Extraction Rules

Types of Goal Extraction Rules:

- Ignorance rule
- Navigation rule (in prev example 😊)
- Relationship rule
- Splitting rule

See paper for details
- Polishing is needed to:
  - Ensure artifact specifications are (grammatically) error-free
  - Ensure artifact specifications conform to our requirements *boilerplates*

- Common errors:
  - “*to improve the quality of travel planning*”
  - “*facilitating the communication between traveller*”
  - Passive voice
  - Adjective-preposition phrase: “Users shall be capable *of creating reviews*”
“Users shall be capable of creating reviews”
Step5. Artifact Classification

- Need to classify types of goals etc

- Extended the Mallet Classifier
  - Machine-learning based general text classifier
  - Extended Mallet to improve its accuracy by 5%
  - Train the classifier with data to classify artifacts according to our defined types (functional feature goal, functional service goal...)
- Construct a goal-use case integrated model based on previous steps – represented by Functional Grammar

- Report problems identified through the extraction
  - A single sentence matches multiple rules, resulting in alternative models
  - An artifact can be classified into multiple types (i.e., has equivalent or close type probability)

- Provide extraction logs (parse trees, dependency trees, artifact classification results, rules applied...
Current Results (see paper for details)

- Over 250 extraction rules
- Over 140 modifying rules
- Evaluated with 2 SRS documents (25+ pages each), achieved:
  - Artifact extraction: 86% precision and 84% recall rates
  - Relationship extraction: 85% precision and 77% recall rates
Challenges

- Depends on Stanford parser for syntactic parsing
- Depends on (extended version of) Mallet for artifact classifying
- Extraction rules and modifying rules require understanding of grammatical dependencies
- Relationships between artifacts in different sentences are not identifiable
  - i.e., The system is designed to improve the quality of travel planning. **This** can be achieved by providing tools to…
Future Work

- Semi-automate extraction rules & modifying rules creation
- Develop technique for cross sentence artifact relationship identification
Demo:

Contact Huan Nguyen for code, examples etc.
huannguyen@swin.edu.au
Meta-Model – Artifact Layer

- **refine**
- **constrain**
- **operationalize**
- **composition**
- **sub-type**

*refine* and *constrain* are many-to-many relationships.

*require* and *exclude* relationships can be established between any pair of artifacts.

- composition
- specify
- sub-type
- start from
- reference
- precede
- resume at
Meta-Model – Specification Layer

S1: System shall be secure
- Positioner(system) + Verb(to-be) + Quality(secure) + Tense(present) + Negation(no)

S2: System notifies users immediately when new messages arrive
- Agent(system) + Verb(notify) + Object(users) + Manner(immediately) + Event(Positioner(Head(messages) + Attribute(new)) + Verb(arrive)) + Tense(present) + Negation(no)
- B1. <transitive action verb><object> ((for) <beneficiary>) ([with | to]) <reference> ([in | at | on] <location>) (for <purpose>)

- B2. <transitive action verb><object> ((for) <beneficiary>) ([in | at | on] <location>) (from <source>) (to <destination>) (for <purpose>)

- FF1. (<agent> [shall (not)] shall (not) be able to]) <transitive action verb> <object> (for) <beneficiary> ([in | at | on] <location>) (from <source>) (to <destination>) (<functional manner>)

- FF2. (<agent> [shall (not)] shall (not) be able to]) <transitive action verb> <object> ([with | to]) <reference> (for) <beneficiary> ([in | at | on] <location>) (<functional manner>)