Identification of Abstractions (report on paper)

D3S Seminar 2011-04-07

http://d3s.mff.cuni.cz

Viliam Šimko
simko@d3s.mff.cuni.cz
Motivation

- Related work for the REPROTOOL project
  - Identification of interesting document fragments
  - Classification of document fragments
  - Creation of the domain model

- Source:
  - Gacitua R., Sawyer P., Gervasi V.: On the Effectiveness of Abstraction Identification in Requirements Engineering, 18th IEEE International Requirements Engineering Conference, 2010
Documents and natural language

• **Natural Language** as a primary medium of communication in software projects.

• **Domain expertise** is encoded in documents:
  - Standards
  - Problem descriptions
  - System specifications
  - Interview transcriptions

• **Abstractions Identification**
  - where domain experts and requirements engineers meet
  - Automatic identification and classification is important
    - especially for large knowledge bases (to lower cognitive load)
What is an Abstraction?

Something in the domain that:
- needs to be examined in the course of analysis
- or whose behaviour and relationships need to be defined as part of the requirements
- or helps to describe what the stakeholders' goals are
Why abstraction identification?

- Organized abstractions can serve a number of useful purposes:
  - The process of identification helps to understand the problem domain
  - Lexicon of terms (project dictionary)
  - Checking the coverage of requirements
  - Context and meaning

---

true fans. "They don't sit on their hands, they actually use them to claggers seem content to sit on their hands until after the election. Lex Ss the film industry will not sit on its hands while the extra-terrestrial chau they would prefer to sit and hold hands with their boyfriend. What the p while the Government sits on its hands and does nothing." >Jim Th E. 19 Viewpoint: World sits on its hands as Rwanda bleeds to death with two children, who sits with her hands folded on her lap throughout if the British government sits on its hands" he couldn't see the ceasefire since March. sits immobile, his hands in prayer, his fleshy underchile the rest of the world sits on its hands over Rwanda at least the Fr. ill our arguments and yet sits on its hands." The MMC said higher price emity. If Mr Major simply sits on his hands through Maastricht 2, there mayor, Jeffrey Higgins, "sits on his hands. "We made a conscious de e, with official Labour sitting on its hands and delivering the Govermen ly, in your coverage, sitting on your hands and reporting what the Gover thing. Among those sitting on their hands are the Canadian Imperial Ba
Defining the problem

- The domain contains abstractions
- A domain document contains terms that act as signifiers of (some of the) abstractions
- The aim is to automatically identify that subset of terms within a document that signify genuine domain abstractions
  - Helping the human analyst in this tedious task
- Using **Automatic Term Recognition.**
Abstractions vs Terms

- In general:
  - Not all terms in a document signify domain abstractions
  - **Most** terms don’t signify domain abstractions
  - Not all domain abstractions are signified by terms in a document
  - **Most** domain abstractions are signified by a term
    - … if we have chosen our document well
- In **most** cases **Automatic Term Recognition** is a good proxy for finding abstractions

- **T** (term) may be relevant even if the abstraction does not occur in the document
- Multiple terms may signify a single abstraction
- **A** (abstraction) is the essence of the document

- **Barcode**, **RFID**, **system repository container** are examples of terms.
Relevance-driven Abstraction Identification (RAI)

- What makes ATR hard?
  - Terms are not the same as words:
    - Acronyms (e.g. “RF”)
    - Multi-word terms (e.g. “RFID tag”)
  - Eventually, terms must be validated by humans

- **Use frequency profiling:**
  - Compare frequency of terms in the domain document and in general language use (normative corpus)
  - Derive log-likelihood (LL) measure of deviation
  - This is a well-established technique
    - …. but mostly it works for single words only
Corpus-based frequency profiling

\( w \): word to be tested for significance
\( D \): domain document (small corpus)
\( C \): normative corpus (large corpus)
\( n_d \): number of words in \( D \) (size of corpus \( D \))
\( n_c \): number of words in \( C \) (size of corpus \( C \))
\( w_d \): number of occurrences of \( w \) in \( D \)
\( w_c \): number of occurrences of \( w \) in \( C \)

We are interested in the significance of a word in the domain document.

\[
E_d = \frac{n_d(w_d + w_c)}{(n_d + n_c)} \quad E_c = \frac{n_c(w_d + w_c)}{(n_d + n_c)}
\]

\[
LL_w = 2 \left( w_d \log \left( \frac{w_d}{E_d} \right) + w_c \log \left( \frac{w_c}{E_c} \right) \right)
\]

- Works well for single-word terms. (used by WMatrix corpus analysis and comparison tool)
- Does not work for multi-word terms
- **Problem:** In specialized domains over 85% of terms are multi-word units
Dealing with multi-word terms

- Multi-word terms can be identified easily, however, we also want to rank the terms.
- Multi-word terms usually occur infrequently in corpora.
- We need to modify the ranking function.

- Using a simple **heuristic** based on number of words in the term.
- **Simplifying assumption** valid for English - last word is important.

\[ S = \sum_i K_i LL_i \]

\[ K_i = \begin{cases} 
  l=1 & \rightarrow & k_0 = 0.1 \\
  l=2 & \rightarrow & k_1 = 0.4, k_0 = 0.6 \\
  l=3 & \rightarrow & k_2 = 0.2, k_1 = 0.3, k_0 = 0.5 \\
  l=4 & \rightarrow & k_3 = 0.2, k_2 = 0.2, k_1 = 0.3, k_0 = 0.5 \\
  l>4 & \rightarrow & k_i = 0, k_3 = 0.2, k_2 = 0.2, k_1 = 0.3, k_0 = 0.5 
\]
RAI algorithm

- Preprocessing
  - POS tagging
  - Stop words removed
  - Lemmatization

- Absolute significance inference
  - LL value computed for every word (using British National Corpus)
  - Multi-word terms are identified (using syntactic patterns)
  - Significance score $S$ is computed for every term

- Relative significance inference
  - Terms are ranked by $S$
Evaluation

- A domain document and gold standard reference
  - 595 pages, 156,028 words
  - An analytical index of 911 items
- Cons:
  - It isn’t a requirements document
- Pros:
  - It is a plausible domain document
  - It’s big
  - It’s bias-free
- Performance of RAI compared to AbstFinder in unsupervised mode

---
Results

• Absolute performance of RAI:
  ▪ 163 terms from 911 matched the index abstractions (if at least 85% similar)
  ▪ Recall/Precision: ~18%

• Relative performance: RAI vs AbstFinder
  ▪ Comparison limited to 200 items
  ▪ Multi-word terms heuristics performs 2x better than the single-word AbstFinder
Relative performance: AbstFinder vs RIA

Relative performance is limited to 200 items

RAI vs AbstFinder - **Recall**

RAI vs AbstFinder - **Precision**

How many of the relevant abstractions could be extracted

How many of the extracted abstractions were relevant
Conclusion

- RAI shows how the significance of multi-word terms can be inferred.
- An example of an unbiased and repeatable evaluation methodology
- A fully-unsupervised mode yields poor results
- Possible improvements:
  - Use iterative semi-automatic approach
  - Take synonyms into account (e.g. using WordNet)
  - Use domain-specific normative corpus