Using Natural Language Parsing (NLP) for Automated Requirements Quality Analysis

Chris Ritter, F. David Ayhan
Why are Quality Requirements Important?
Cost Overrun vs Systems Engineering Effort

NASA found a smaller investments in systems engineering proportional to cost overrun

*Source: Werner M. Gruhl. Chief Cost & Economics Analysis Branch, NASA Headquarters*
Cost of Error in Requirements Phase

Expect a **10 to 1000x increase** in cost for errors missed in the requirements/design phase

(“Extra Time Saves Money”, Warren Kuffel)
Where are my Requirements?
Why not Microsoft Office?

- **Many Versions** - Often documents are emailed around and managing consistency is difficult
- **Change Tracking** - Change logs can be kept manually but often finding the rationale for a requirement change is impossible
- **Traceability** - Maintaining traceability between requirements and artifacts becomes a costly and manual process
Why start with models?

• Problem:
  – Natural language requirements can be difficult for humans to read and are often ambiguous

• Current Solution:
  – Models can be executed in a simulation environment for verification and validation
  – Models can generate consistent documentation
  – Models can automatically generate necessary views (requirements document, hierarchy chart, requirements diagram etc.)

• New Problem
  – Models almost always become documents
  – How do we ensure quality of natural language documents?
What Makes a Good Requirement?
Attributes of Quality Requirements

- **Correct** – i.e., describes the user’s intent and legally possible
- **Complete** – i.e., expresses a whole, single idea, and not portions of one or many
- **Clear** – i.e., explicit and not confusing to readers
- **Consistent** – i.e., does not conflict with other requirements
- **Verifiable** – i.e., proves within realistic cost and schedule that the architecture meets the requirement
Attributes of Quality Requirements

- **Traceable** – i.e., uniquely identify, and able to be tracked to predecessor and successor lifecycle items/objects, such as functions or components.
- **Feasible** – i.e., implement with existing or projected technology and within cost and schedule.
- **Modular** – i.e., changes without excessive impact on other requirements.
- **Design** – i.e., does not impose a specific solution ("what" not "how").
Requirement Pitfalls

• DON’T use:
  – ambiguous language
  – use bullet lists; use numbered lists instead
  – jargon
  – language that provides an escape clause Ex: “The user shall be able to access the Internet as often as is practicable”
  – write long, rambling sentences
  – two requirements in one sentence; e.g., “The system shall … and …”
  – vague terms -- Ex: “user-friendly”
  – include suggestions or possibilities – Ex: “may”, “should”, “ought”
  – wishful thinking – Ex: “The system shall be 100% reliable”
Natural Language Processing

• Definition: “a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages”

• Innoslate uses this technology to break down sentences into nouns, verbs, and adjectives to identify when conjunctions are used (clear), specific types of hardware/software specified (design), and other parameters that affect the requirement’s quality
Part of Speech Tagging

*Given:*  
The system shall identify the part based on either a Serial Number or a Part Name.

*Tagged as:*  
The (Determiner) system (Noun) shall (Modal) identify (Verb) the part based on either a Serial Number (Noun) or a Part Name (Noun).
Given:
The system shall start within 5 seconds but can take 10 seconds.
Complete - Sentence Detection

*Given:* The system shall start within 5 seconds
Complete - Run-On Detection

**Given:**
The system shall identify the part based on **either a Serial Number Part Name** that are also acceptable input.
Design - Predetermined Solution

Given:
The system shall use Microsoft SQL Server 2008R2.
Verifiable - Ambiguous Words

Given:
The system shall be very fast.
Coming Soon

- Fragment Detection
- Active Voice Detection
- Multiple Negative Detection
- Exact Duplicate Identification
- Near Duplicate Identification
Next Generation

• Utilize machine learning to develop self-improving algorithms
  – Gather large data sets of requirements
  – Process requirements with neural networks
  – Analyze network to identify good patterns and common pitfalls
Contact Information

• Christopher Ritter
• 301-910-1818
• SPEC Innovations
• chris.ritter@specinnovations.com