Foundations for an Expert System in Domain-Specific Traceability

Jin Guo, Jane Cleland-Huang, Brian Berenbach

Contents

- Background
- Related Work
- Overview of the System
- Components of the System
  - Constructing Knowledge Base
  - Mapping to Linguistic Model
  - Inference Rules
- Experiments
- Future Work

Contents

Background

- Software Traceability
  - To demonstrate that all identified hazards have been sufficiently mitigated in the delivered system, and that the system is safe for use.
- Trace links
  - incrementally constructed between hazards, contributing faults, mitigating requirements, design, code, and test cases. [1]

Background

- Current techniques for trace creation and maintenance
  - VSM, LSI, probabilistic network, etc. [2][3][4]
  - Problems: low precision rate
- Barrier:
  - mismatch of terminology
  - Example :
    - *Highway Wayside Segment shall monitor signal, road work directive, and hazard detector information from field devices.*
    - *During lamp-out conditions the WIU shall send the current state of the highway signal.*

Contents

Related Work

- Evaluating regulatory compliance of product requirements to standards(T. D. Breaux, A. I. Antón, et al) [5][6][7]
- Creating and mapping ontologies from requirements (L. Kof, N. Assawamekin, et al.) [8][9]
- Generating for traceability relations using Rule-based approach (G. Spanoudakis, A. Zisman, et al) [10]

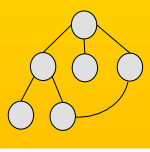
Contents

Overview of the DoCET System

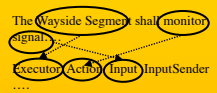
- Domain-Contextualized Expert Traceability system
  - Knowledge Base
  - Linguistic Model
  - Inference Rules

Overview of the DoCET System

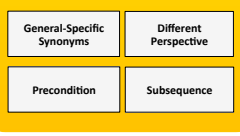
❶ Construct or acquire a domain-specific knowledge base of concepts and relationships.



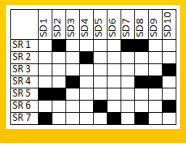
❷ Map requirements to a format interpretable by an inference engine.



❸ Construct inference rules that depict general heuristics for retrieving expected requirement links.



❹ Generate trace links using the constructed expert system.



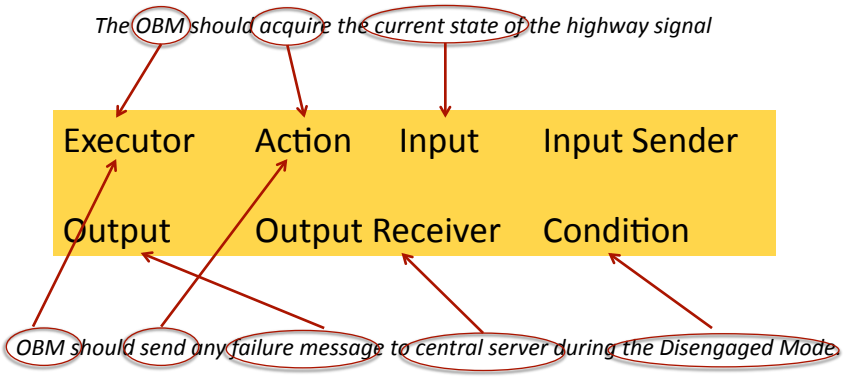
Contents

- Background
- Related Work
- Overview of the System
- **Components of the System**
  - **Constructing Knowledge Base**
  - **Mapping to Linguistic Model**
  - **Inference Rules**
- Experiments
- Future Work

Constructing Knowledge Base

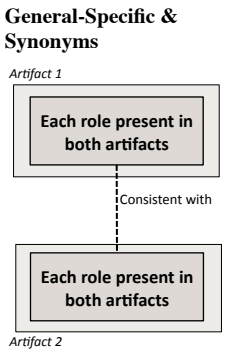
- **Prolog**: logical programming language
- **Vocabulary**: object, object type, attribute, relationship
  wayside_segment, system, operational, synonym, …
- **Basic Facts**: atomic sentence.
  is_a(wayside_segment, subsystem)
- **Complex Facts**: complex formulas
  is a(X, self_diagnostic) :-
  is a(X, diagnostic),
  X \= on_demand_diagnostic.
- **Terminological Facts**: Disjointness, Symmetry, Inverses, etc.
  father_of(X, Y) :-
  child_of(Y, X).

Mapping to Linguistic Model

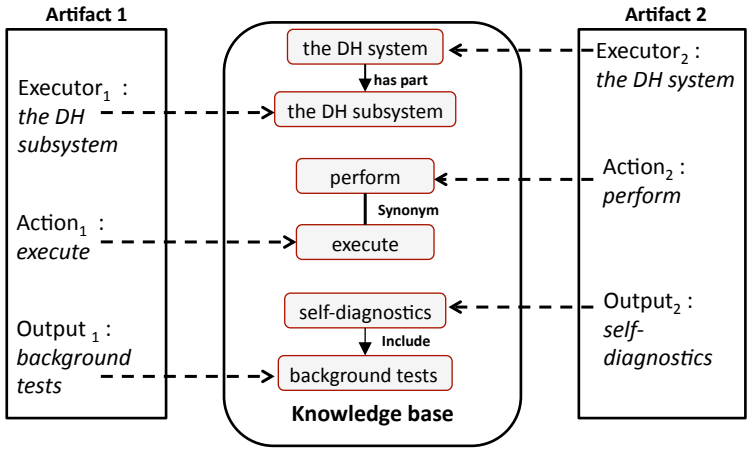


Inference Rules

- Rule1&2: General-Specific, Synonyms
  - Artifact 1: *the DH subsystems shall automatically execute background tests for critical functionality without impact on current operation.*
  - Artifact 2: *the DH system shall be capable of performing self-diagnostics.*

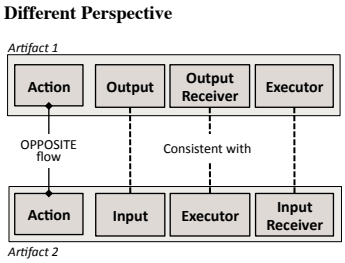


Inference Rules



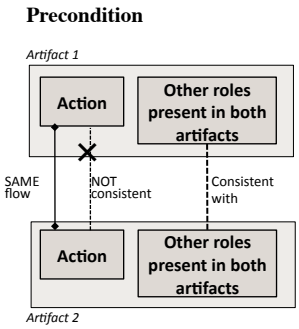
Inference Rules

- Rule3: Different Perspective
  - Artifact 1: *Automobile segment shall send the highway signal to the central control system.*
  - Artifact 2: *The control segment shall receive information from the automobile segment.*



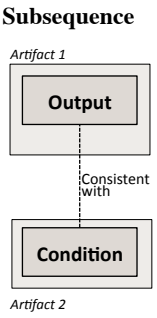
Inference Rules

- Rule4: Precondition
  - Artifact 1: *The DH system shall provide the ability of each subsystem to upload diagnostic data to the HCS.*
  - Artifact 2: *The DH system shall provide centralized logging of diagnostic information..*



Inference Rules

- Rule5: Subsequence
  - Artifact 1: *The system shall enter Initializing Mode at the start of a new mission, and occupy this mode until road database update and departure test procedures have completed successfully.*
  - Artifact 2: *Upon completion of the initialization tasks, the OBM shall transition into Disengaged Mode.*



Contents

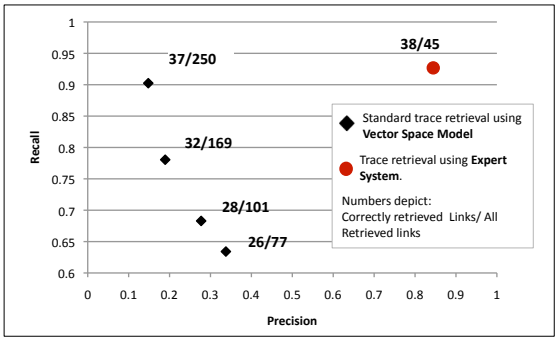
- Background
- Related Work
- Overview of the System
- Components of the System
  - Constructing Knowledge Base
  - Mapping to Linguistic Model
  - Inference Rules
- **Experiments**
- Future Work

Experiments

- Datasets
  - 224 System Requirements (SRs),
  - 945 System Design artifacts (SDs)
  - 582 SubSystem Requirements (SSRs).
- Initial Focus
  - small subset of functional requirements that included 30 SRs, and 24 SDs, 41/720 correct links.

Experiments

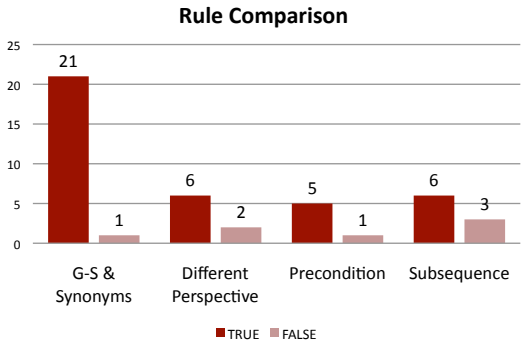
- Experiment 1: Does the Expert System improve traceability?
  - Our approach successfully identified 38 out of 41 true links, and only retrieved 7 false positives resulting in recall of 0.93, precision of 0.84, and F2-Measure of 0.898.



Experiments

- Experiment 2: Are all five inference rules effective?

Rule 1,2: 55%, 0.955
Rule 3: 16%, 0.75
Rule 4: 13%, 0.833
Rule 5: 16%, 0.667



Experiments

- Experiment 3: Are the inference rules generalizable to other domains?
  - Ten traceability links between requirements for the World Vista electronic health record and health information system and certification requirements taken from the Certification Commission for Health Information Technology (CCHIT).
  - Rule General-Specific: 6/10, Different Perspective: 3/10

Experiment

- Experiment 4: Is an initial domain model effective for tracing additional requirements?
  - 10 new SDs, 24 SRs: 14 true links
  - Without additional knowledge: expert system returns 5 true links and 1 false link.
  - With 9 items of additional knowledge: expert system returns 13 true links and 1 false link.

synonym(field_device, field _element)
synonym(broadcast, send)
synonym(automobile_segment, car)
…

Contents

- Background
- Related Work
- Overview of the System
- Components of the System
  - Constructing Knowledge Base
  - Mapping to Linguistic Model
  - Inference Rules
- Experiments
- **Future Work**

Future Work

- Automate the KB creation and the mapping from artifacts to the linguistic model
- Construct expert system across multiple domain, expand and fine-tune the identified rules
- Expand the expert system to support non-functional requirements.

Reference

- [1] Jane Cleland-Huang, M. Heimdahl, J. Huffman Hayes, R. Lutz, and P. Maeder. Trace queries for safety requirements in high assurance systems. In *Intn'l Working Conf. on Requirements Eng.: Foundation for Software Quality, 2012*.
- [2] G. Antoniol, G. Canfora, G. Casazza, A. De Lucia, and E. Merlo. Recovering traceability links between code and documentation. *IEEE Trans. Softw. Eng., 28(10):970–983, 2002*.
- [3] A. De Lucia, F. Fasano, R. Oliveto, and G. Tortora. Enhancing an artefact management system with traceability recovery features. In *International Conference on Software Maintenance, pages 306–315, Washington, DC, USA, 2004*. IEEE Computer Society.
- [4] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram. Advancing candidate link generation for requirements tracing: The study of methods. *IEEE Trans. Softw. Eng., 32(1):4–19, 2006*.
- [5] T. D. Breaux and D. Gordon. Regulatory requirements traceability and analysis using semi-formal specifications. In *Intn'l Working Conf. on Requirements Eng.: Foundation for Software Quality, 2013*.
- [6] T. D. Breaux, A. I. Ant´on, K. Boucher, and M. Dorfman. Legal requirements, compliance and practice: An industry case study in accessibility. In *RE, pages 43–52, 2008*.
- [7] T. D. Breaux, A. I. Ant´on, and J. Doyle. Semantic parameterization: A process for modeling domain descriptions. *ACM Trans. Softw. Eng. Methodol., 18(2):5:1–5:27, Nov. 2008*.
- [8] L. Kof, R. Gacitua, M. Rouncefield, and P. Sawyer. Concept mapping as a means of requirements tracing. In *MaRK'10), 2010*.
- [9] N. Assawamekin, T. Sunetnanta, and C. Pluempitiwiriyawej. Ontologybased multiperspective requirements traceability framework. *Knowl. Inf.Syst., 25(3):493–522, 2010*.
- [10] G. Spanoudakis, A. Zisman, E. P´erez-Mi˜nana, and P. Krause. Rule-based generation of requirements traceability relations. Journal of Systems and Software, 72(*2):105–127, 2004*.

Discussion Time

Thank you