# A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System

**Abstract:**

The efficiency and effectiveness of the recovery of traceability links in requirements management is becoming increasingly important within interdisciplinary industry. Due to the complexity of production development such as automotive, software industry and the aerospace industry, managing requirements is indispensable and challenging. Products in these industries are constantly being updated and modified as the understanding of risks increases with experience and new products are developed in light of such risk. Therefore, the traceability links among the requirements artifacts, which fulfill business objectives, is so critical to reducing the risk and ensuring the success of products. To that end, this paper propose a semantic based traceability link recovery (STLR) architecture. According to best of our knowledge this is the first architectural approach that uses DBpedia knowledge-base and Bablenet 2.5 multilingual dictionary and semantic network for finding similarity among requirements and the automation of the recovery traceability links using our novel Triple extraction, and Triple disambiguation algorithm. Our preliminary results show the effectiveness in term of precision and recall compared to Vector Space Model and Wu Palmer algorithm.

SECTION I.

## Introduction

In the previous generation of National Air System (NAS) the collaboration among systems were point-to-point, application level data interface, and predefine agreement. To increase the efficiency of Air Traffic management system a new generation of NAS needed that relies on a “System Wide Information Management (SWIM)” [1]. Building SWIM success depends on robust Requirements Management system (RMS) which assure the interoperability, meeting the standards architecture of SWIM, and security. Further, SWIM provides governance to NAS programs which is the ability to ensure that all of the independent stakeholders’ efforts (whether in the design, development, deployment, or operations of a service) come together to meet enterprise requirements. To that end, a robust RMS has to have an effective traceability link recovery framework to define the relatedness, to trace the changes, and to understand the impact of changes among SWIM requirement artifacts. As a requirements management expert, Peter Zielczynski defines the major steps in requirements management as follows: establishing a plan, eliciting requirements, developing the Vision document, creating use cases, supplementary specification, and finally a system design see Figure 1. Indeed, the backbone of requirements management is “requirements traceability” by which it is possible to map individual artifacts of requirements with other artifacts in the system. Furthermore, the requirements traceability plays a vital role in ensuring the current requirements are met, traces the impact of such changes to the requirements, and defines the relationship between them and a delivered system which, in turn, ultimately lowers risk. Essentially, what is at stake here is the traceability of the requirements through a manual process of mapping and detecting semantic similarity between artifacts. Moreover, time consumption is kept in mind. To fulfill these requirements, the framework that we propose here offers specific services for the management of requirements. It ensures that projects meet their objectives and business expectations by the automating the recovery of the traceability links process and using a semantic web technologies for generating the recovery of those links with high accuracy.

SECTION II.

## Related Works

Not much work has been reported in literature on tractability link recovery using semantic technologies. Additionally, according to best of our knowledge, no such initiative has been taken particularly for the domain of aerospace. The research on Automated Traceability Link Recovery [2] focuses mostly on software engineering life cycle using Information Retrieval techniques (IR) [3]. IR is mainly defined as discovering processes of nominee traceability links on the basis of the similarity between software engineering artifacts. Mahmoud et al. [4] proposed Wikipedia-based semantic relatedness approach called Explicit Semantic Analysis. The limitation of this approach is use of Wikipedia, a not trusted source for judgment’ and it is only valid for software engineering life cycle, and does not consider any industrial business models. Antoniol et al. [5] proposed a technique to automate the recovery of traceability links by finding the similarity of pair property from object oriented design and code. This approach assumes that the relations among properties are maintained during the transformation of design into code and this assumption itself is one of the limitations. In practice, the code does reflect the design but did not necessarily maintain the similarity. Also, the process of selecting the properties that measure the similarity is a predefined process. Marcus et al. [6] proposed using recovery of traceability links by using Latent Semantic Indexing. However, this proposal uses non-proactive approach does not make use of word order, syntactic relations, and morphology. In addition, two documents written in different languages will not appear similar. Antoniol et al. [7], in their approach use vector space mode and this approach does not take account relations among terms of documents.

SECTION III.

## Application

One of its practical uses, for instance, is providing weather services for air traffic aviation in Detroit. As long as these services are compliant with the System Wide Information (SWIM) standards, the provider should ensure the application follows the Federal Aviation Administration's (FAA) governance policies and standards for producing and consuming data products compliant with SWIM. To that end, the provider creates links between the requirement and the standards to ensure that all of the independent stakeholders’ efforts in the Software Development Life Cycle agree with FAA's policies and standards. This process is extremely costly, however, in terms of both the consumption of time and resources. Yet we believe that our proposed A Semantic Approach for Traceability Link Recovery in Aerospace Requirements Management System (STLR) can be used to reduce those costs and increase the effective and efficiency of the creation of traceability links. Moreover, our approach can be used by product owners, product managers, business analysts, and developers alike.

### A. Features

The proposed approach comes with many features that can be listed as follows: (1) generating traceability link candidates, (2) generating traceability link matrices, (3) generating traceability link diagrams.

### B. Application Requirements

The STLR framework shall support automated recovery of traceability links with high accuracy that are measured in terms of precision and recall, without manual input from human beings. In addition, the user shall be able to filter and control the threshold of the traceability link candidates.

SECTION IV.

## Preliminary

### A. BabelNet

BabelNet [8] is a multilingual encyclopedic dictionary and utilizes different semantic knowledge-base, namely WordNet, Open Multilingual WordNet, Wikipedia, OmegaWik, Wiktionary, and Wikidata. In addition, it is semantic network which connects concepts and named entities in a very large network of semantic relations.

### B. Dbpedia

DBPedia [9], in general, is data set extracted from Wikipedia and transformed to Resource Description Framework (RDF) model data, using the Notation3 (N3) data representation format. In addition, DBPedia allows to run sophisticated queries against Wikipedia, and to link the different data sets on the Web to Wikipedia data

SECTION V.

## Semantic Based Tractability Link Recovery Architecture (STLR)

Due to the complexity of the product development within diverse industries, it is imperative that the recovery of traceability links in requirements management be efficient and effective. We propose a solution that uses a recovery flow of traceability links with variations in its implementation from existing models in the literature. Figure 2 shows the system architecture of components. Our architecture is comprised of five components. viz: “Natural Language Processing” (NLP), “Triplet Extraction”, “Disambiguation Process”, “Category Extraction” and “Inference Engine”.

### A. NLP Module

The NLP module uses GATE framework [13] for uploading the artifacts, tokenization, stemming, and lemmatization [10]. Tokenization is the process that splits the artifacts into tokens, where stemming and lemmatization are the process of converting or removing the inflexional, derivational form to a common world form. Furthermore, the module exploits Stanford Parser specifically LexicalizedParser function to construct a structure tree that can be characterized each node as a noun phrase, verb phrase, or a full stop (.) and generates a syntactical structure of sentences. After that we split the constructed tree into subtrees based on specified tag which is‘S’ in our implementation see Algorithm 1.

### B. Triple Extraction Module

The important task of this module is extraction of Triplets: the subject, the predicate, and the object from the document. The subject is the resource that is being described by the ensuring predicate and object. The Object is either a resource referred to by the predicate or a literal value. The predicate is a relation between the subject and the object. Our proposed algorithm is an extension of “Triplet Extraction from Sentences” [11] algorithm to cater a) the multiple subjects and objects in one sentence b) to consider multi-word (e.g. Information Technology) *as* subject and/or object. The process of this module is as follows:

#### 1) Subject Identification

To find the subject, we use the Noun Phrase (NP) sub-tree. Since the NP might have compound subjects, thus we search for all subjects NP sub-tree. Otherwise, we search for all preposition, else we search for all adverbs and WH-adverbs. Algorithm 2 shows the pseudo code of the algorithm

#### 2) Predicate Identification

We get the Verb Phrases (VP) from the structured tree of the NLP module, and then the predicate is extracted from the deepest verb in the VP tree. Algorithm 3 shows the pseudo code of the algorithm

#### 3) Object Identification

We get the Verb Phrases (VP) from the structured tree of the NLP module, then finding the object is accomplished by finding one of the nouns else preposition, and adjectives else by past tense verbs in VP tree. Algorithm 4 shows the pseudo code of the algorithm.

SECTION C.

## Triple Disambiguation Module

The cornerstone issue of natural language processing is Word Sense Disambiguation (WSD) which is the task of identifying the correct meaning of the word or phrase within the context. In our approach we introduce a novel technique to identify which sense of the polysemous subject or object should be used in particular contexts by first fetching the senses and glosses of both from Babelnet. Second, Disambiguation words algorithm is applied on the glosses which returns disambiguated senses of subjects predicates and objects. If document does not contain enough text, enriched the document by utilizing knowledge available in the babelnet. Document Enrichment: Following steps have been followed in order to enriched the document: 1- Get the subjects and objects glosses of the triples from Babelnet. 2- Send these glosses to the first module of the system which is NLP module and then follow the flow of the system. Document enrichment helps us to improve an accuracy in results.

This module has a basic task: to accept the DBpedia category from Disambiguation module for each sense, finding the super-category and subcategory of the input category from DBpedia, and fetching DBpedia categories for each disambiguated sense see Figure 3. Algorithm 5 shows the pseudo code of the algorithm

### D. Inference Module

The Inference module functionality maintains a count for the Overlapping of DBpedia categories and tokens between two artifacts and the calculation of a ‘similarity score’. In addition, it generates candidates for traceability links. Assuming I > J > K our weighting scheme as follow:

1. If one subject or object category from document DBpedia Category matches with the other document DBpedia category then add + I weightage to the count.
2. If one subject or object category from document sub Category match with the other document sub category then add +1 weightage to the count.
3. If one subject or object category from document super Category match with the other document super category then add + K weightage to the count.
4. If one subject or object category from document DB-pedia Category match with the other document sub category or super category then add + K weightage to the count.
5. If one subject or object category from document Super Category match with the other document sub category or DBpedia category then add +K weightage to the count.
6. If one subject or object category from document Sub Category match with the other document super category or DBpedia category then add +K weightage to the count.
7. If similar token encountered in the documents and have same sense then add +1 weightage to the count.

SECTION VI.

## Evaluation

In order to measure the efficiency and the accuracy of the proposed framework, we implemented an experimental framework using General Architecture and Engineering Text (Gate) which is a development environment that provides a rich set of interactive tools for the creation, measurement and maintenance of software components for processing human language, Stanford NLP APls which is a program that works out the grammatical structure of sentences by using the parser in question. Each sentence of the document is parsed in the tree structure [10]. Ultimately, in order to simulate a real world scenario we selected a data set of requirements management tools from Borland CaliberRM [14].

### A. Results

In the experiment, the accuracy of our approach framework is compared against Support Vector Model [15] and Wu Palmer algorithm [16]. The primary accuracy measures used for comparison are Precision and Recall. The comparison was carried out as follows:

1. Uploading two sets of requirement artifacts
2. Setting the threshold
3. Reading requirement by requirement from the first set
4. Looping through each requirement in set two
5. Calculating the semantic similarity between set one requirements and set two requirements
6. Recording precision and recall

First of all, a general comparison of the models points that our approach heuristic accomplished steadily better result than the benchmark algorithms in terms of precision, however it matches VSM in term of recall in threshold equal 0.05.

Figure 4 shows the precision chart graph for Precision measurement. Precision is the ratio of the number of true positive links retrieved over the total number of inks retrieved.

Figure 5 shows the recall chart graph for Recall measurement. Recall is the ratio of the number of true positive links retrieved over the total number of true positive links.

SECTION VII.

## Future Work

In pursuance of this solution, we will evaluate large datasets from the Air Traffic Aviation system. Furthermore, we will create Web-based applications that are compliant with the SWIM standards and which we believe in agreement with the governance process of the SWIM program. Finally, we are committed to optimizing the algorithm by improving its precision and recall.

SECTION VIII.

## Conclusion

These inter-industrial projects are constantly updated and modified in light of new risks and developing products. Traceability links are a vital part of requirements managements for these industries. Since Automated Traceability Links are an important element to ensure the success of Software engineering projects, our proposed framework helps Software engineer projects to meet business requirements, to improve the precision and recall of traceability links between requirements artifacts, and increases the efficiency of time management.