Towards More Intelligent Trace Retrieval Algorithms

Jane Cleland-Huang and Jin Guo
Systems and Requirements Engineering Center
DePaul University
Chicago, IL USA
jhuang@cs.depaul.edu; jguo9@mail.depaul.edu

ABSTRACT

Automated trace creation techniques are based on a variety of algorithms ranging from basic term matching approaches to more sophisticated expert systems. In this position paper we propose a classification scheme for categorizing the intelligence level of automated traceability techniques. We show that the vast majority of relevant work in the past decade has been focused at the lowest level of the Traceability Intelligence Quotient (tIQ) and posit that achieving high quality automated traceability will require re-focusing research efforts on the development of more intelligent algorithms capable of reasoning about concepts, their relationships and constraints, and the contexts in which they occur.

Categories and Subject Descriptors

D.2.1 [**Software Engineering**]: Requirements/Specifications; 1.2.5 [**Artificial Intelligence**]: Programming Languages and Software—*Expert System Tools and Techniques*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing — *Text Analysis*

General Terms

Documentation

Keywords

Traceability, Expert System, Domain Ontology

1. INTRODUCTION

Software traceability is an essential element of the software development process, especially in safety critical systems where it plays an integral role in demonstrating that a system is safe for use [5, 21]. For example, regulatory standards such as the US Federal Aviation Authority's (FAA) D0178b/c [10] require trace paths to be established between identified hazards, contributing faults, mitigating requirements, design, code, and test cases. Furthermore, in current fast-paced incremental and/or continuous integration life-cycle processes, traceability serves the useful purpose of helping developers to identify parts of the code to be changed or to predict the cost and impact of proposed changes. Traditional approaches to traceability, including those integrated into most commercial tools, assume that users will create and maintain trace links manually. Unfortunately, the tracing task is arduous and error-prone and therefore in practice even the most basic trace links are often inaccurate, incomplete, and ambiguous [24, 28].

RAISE'14, June 3, 2014, Hyderabad, India
ACM 978-1-4503-2846-3/14/06
http://dx.doi.org/10.1145/2593801.2593802

1.1 Automated Trace Creation

To address these problems, numerous researchers over the past decade have investigated and developed novel approaches for automating the generation of trace links [1, 6, 16]. The basic idea is to use information retrieval techniques to compute similarity scores (or link probabilities) between pairs of source and target artifacts, such as between requirements and code, or between regulations and requirements, and then to present high-scoring pairs as candidate links to users. The quality of the resulting trace links are measured using standard metrics such as recall (fraction of correct links retrieved), precision (fraction of retrieved links that are correct), and other metrics such as Mean Average Precision (MAP) which measures the extent to which correct links are placed towards the top of a ranked list of candidate links. Results have been mixed, typically returning maximum recall values of about 90% and precision ranging from about 20 to 40% [19]. Not surprisingly, practitioners have deemed these results inadequate for use in industry, and as a result, trace generation techniques have not yet been adopted broadly in practice.

We posit that the core limitation of existing traceability results, and the reason we have not yet achieved our goal of industrial strength automated traceability, is that the algorithms we have adopted lack sufficient intelligence to handle the complexities of real world requirements [14].

1.2 How Humans Create Links

Through observing practitioners as they create trace links we can see that the task is quite complex [25]. For example, in a typical scenario the trace analyst might first look for direct matches and/or synonyms between terms in the source and target artifacts. She might then identify and reason about higher level relationships between concepts such as *is-a* or *comprised of* associations to find possible matches. Once a potential match has been found, the analyst then evaluates it within the context of pre-conditions, post-conditions, and other kinds of exclusionary clauses. The

task is complicated by the fact that traceable artifacts are often described using legal or domain-specific terminology, and using complex sentence structures.

For example the following high level design requirement is taken from NASA's publicly available CM1 dataset:
"*The DPU-TIS shall report errors to an application program by setting the ERRNO task variable and returning ERROR.*"

The requirement specifies the constraint that errors must be reported in a certain manner by setting the ERRNO task variable and by returning ERROR. Performing a trace retrieval query on the CM1 low level design requirements returns many artifacts which refer in any general way to the topic of reporting errors. The following is one such example of an incorrect result.
"*Each time it wakes, ccmTask checks to see if it is time to form an error/event packet for transmission to the ground. If so, ccmTask calls ccmHkMkError to actually create the packet and forward it to DPU SCUI for transmission to the ground.*"

While the low-level requirement does refer to errors and to the DPU, and is therefore included in the list of candidate links, it is actually incorrect because its focus is on transmitting error event packets to the ground, rather than on setting a task variable. We provide this example to illustrate the complexity of the traceability problem.

1.3 Towards More Intelligent Trace Creation

Focusing purely on term matching techniques is therefore unlikely to produce a precise and complete set of trace links, simply because it is an oversimplification of the cognitive steps that need to go into the trace creation process.

In this paper, we propose a classification scheme which identifies various degrees of intelligence exhibited by text-centric trace creation algorithms and posit that continued efforts to achieve high degrees of automation will require focusing our efforts on developing more intelligent tracing solutions. We choose to focus purely on text-based techniques because these are a core component of almost every automated, or semi-automated tracing algorithm.

Our classification scheme therefore excludes traceability techniques that are primarily structural in nature [31], draw on runtime execution traces [8], are process oriented, or based on associations between work items [3]. While these techniques make important contributions to advancing traceability solutions and have been shown to improve the quality of generated trace links, they are typically not stand-alone methods and are often used in conjunction with information retrieval algorithms. We therefore classify such work according to its underlying retrieval approach. More importantly we believe that utilizing such techniques provides additional *evidence* for a link, but does not directly contribute to the intelligence of the underlying algorithm. As such, each of these techniques can be used in conjunction with core trace generation algorithms at any level of our classification scheme.

1.4 Traceability IQ Scale

Our focus is on the reasoning ability of the underlying trace algorithms. Our traceability intelligence quotient (tIQ), summarized in Figure 1, is defined on an ordinal scale that includes the abilities to (1) match artifacts based on the terms they contain, (2) draw on external evidence to build untyped associations, (3) build semantic associations, (4) reason analytically through the use of an expert system, and finally (5) to synthesize knowledge. There is a clear progression in intelligence through the first four classes; however while 'knowledge synthesis' is clearly an important form of intelligence, it does not fit so easily into the hierarchy and can be applied in different ways across a variety of levels. We therefore present it as a cross-cutting concern. We have identified these classes subjectively based on our own observations and intuitions and lay them out in this position paper as an initial starting point for further discussion.

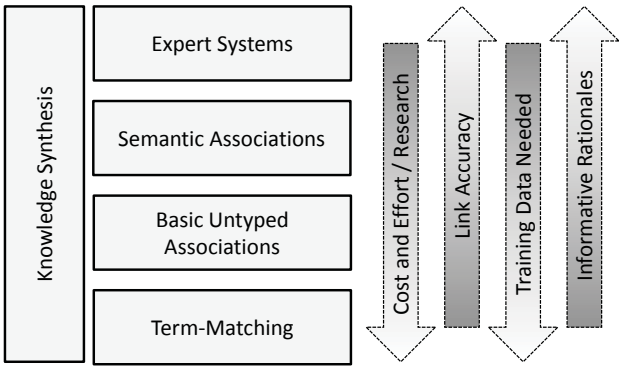

Figure 1: Levels of the Traceability Information Quotient showing hypothetical trends in trace accuracy. Improvements have already been demonstrated as we move from term-matching to basic associations.

In Sections 2 to 6 we describe each of these tIQ categories. In Section 7 we then classify traceability work published in the last decade, between 2004 and 2013, according to the tIQ levels and point out that the majority of effort has been, and continues to be, exerted at the lowest level of the tIQ. Finally in Section 8 we outline future research directions that are needed to transition to more intelligent tracing solutions.

2. TERM MATCHING

The lowest level of intelligence in automated trace generation techniques uses terms, or groups of similar terms, found in the artifacts themselves, to establish trace links. The notion is that trace links can be constructed when terms in the source artifact match those in the target artifact. The most popular approach is based on the Vector Space Model (VSM) [17]. Variants which introduce some degree of improvement in the quality of generated trace links include the use of matching term lists, project glossaries, and other forms of thesauri [17]. Techniques such as Latent Semantic Indexing (LSI) or Latent Direchlet Allocation (LDA) overcome the strict term-to-term matching of the VSM through using processes such as Singular Value Decomposition to discover simple associations between terms by extracting and leveraging the contextual usage of each word. However, we place such algorithms in the term-matching category because their only inputs are the terms in the artifacts.

We also place user feedback techniques such as Rocchio [17] or Direct Query manipulation (DQM) [29] into this category because such techniques leverage user feedback to simply increase or decrease term weights, to add entirely new terms to the trace query, and/or to completely eliminate terms. While such techniques return improvements over the

more basic approaches, all approaches in this tIQ class share the common characteristic that they only work well when source and target artifacts contain a sufficient set of matching terms.

Historically, this limitation has led to fairly significant constraints on the quality of the generated trace links. On the other hand, such approaches can be applied with no prior knowledge and no training requirements.

3. BASIC UNTYPED ASSOCIATIONS

The second category of tIQ algorithms uses external evidence or feedback from multiple trace queries to discover and leverage associations at the artifact level. Techniques that fall into this category are characterized by the fact that they do not understand the semantics behind the association. Several different techniques fall into this class of algorithms.

One example in the traceability domain is based on learning query transformation rules from an initial set of approved trace links [7]. For example, this approach might learn that a source artifact containing terms a and b was traced at some level of confidence and support to target artifacts containing the term c, and then use that information to support future trace queries containing similar terms. This approach is relatively "dumb" in that it does not understand the reason for the association, but just leverages the fact that exists.

Another example uses either structural or term-based clustering to leverage the general idea that groups of similar items tend to be linked to other groups of similar items [26]. In practice this means that if a link is discovered from a source artifact s to target artifact t, and if t is part of a group of artifacts T such that $t \in T$ then this increases the probability that other artifacts in T should also be linked to s. Again, such techniques typically do not try to understand the clusters, but simply leverage the fact that they exist.

Other researchers have used search techniques to either augment concepts in the query with new facts, or to search for concepts which make connections between terms in the source and target artifacts. For example, Gibiec et. al [12] explored the use of query augmentation techniques, which used the source artifact as a search query, then searched for related documents on the internet, analyzed the text to extract a set of representative terms, and used these to augment the text in the source artifact for purposes of issuing a trace query. In effect, this approach dynamically searches for associations between terms in the source and target artifacts. It also fits into our untyped category, because it produces mappings without the ability to explain them. Such techniques are constrained by the availability of relevant and searchable data and therefore work better in commonplace domains for which relevant information is available on the internet.

As a final example, Asuncion et al. instrumented a project environment to monitor project stakeholders' activities in order to identify artifacts such as sets of classes, requirements, and/or test cases which were either modified, viewed, or utilized in succession [3]. Other researchers have explored similar techniques for associating items that are checked in to version control systems or referenced together in the commit log. The underlying notion is that items that are changed together are related and should therefore be associated by trace links. Associations created this way are formed between artifacts that correspond to the granularity of the version control system. While we might infer that a test case *tests* a requirement, the semantic knowledge is typically only used to label the generated trace link and is not used to reason about its creation.

4. SEMANTIC ASSOCIATIONS

The third level of the tIQ utilizes a knowledge base (KB) to create and utilize semantically aware associations in the trace creation process. Such knowledge bases are composed of a set of data that includes basic terms that define the vocabulary of the domain, and a set of sentences that describe the relationships between those terms [18, 13, 30]. The data is often represented in the form of an ontology in which knowledge is constructed using logical operators such as *AND* and *OR*, as well as *implication* and *negation* operators to build complex ideas from more primitive concepts [18].

For traceability purposes, researchers have proposed using ontology to connect concepts in source and target artifacts [15, 2]. For example, term a and term b might be related through an "is-a" relationship in the knowledge base such that $ais - akindofb$, allowing trace links to be created between general and specific instances of a occurring across pairs of artifacts. Various approaches have been proposed which weight the evidence for a trace link according to factors such as the distance between concepts in the ontology [22] and the semantics of their associations.

However, creating useful ontologies is very challenging. Existing ontologies tend to represent everyday objects, or medical and/or scientific terms rather than the engineering concepts needed to trace requirements, regulatory codes, and other kinds of entities in software intensive systems. Ontologies used for traceability purposes must therefore often be custom built to support very specific domains of use, a task which is difficult to fully automate and which is therefore costly to perform.

On the positive side, ontologies can be built once and used often. Furthermore they can be constructed in advance and imbued with rich semantics. If new algorithms can be created that take full advantage of underlying knowledge bases, then investing the time to construct such knowledge bases for use in the tracing process makes sense. This is especially true in safety-critical domains such as medical devices and transportation in which traceability costs alone can grow to millions of dollars [4].

While ontology-based traceability solutions are capable of assisting in the creation of trace links, they address only half of the traceability problem, as they fail to reason about the impact of conditions and constraints upon the final trace link. Our earlier example about error reporting in the CM1 system illustrated this point. Approaches that rely purely on ontology therefore may improve recall but fail to adequately address the precision problem [22]. Current ontology-based techniques are still immature in the traceability domain and have so far led to only minor improvements in the quality of generated trace links.

5. EXPERT SYSTEMS

Our fourth category of tIQ addresses the weaknesses of ontology alone and represents more intelligent approaches that reason about concepts, conditions, and constraints. In general, an expert system emulates the decision-making pro-

cess of a human expert [18]. Expert traceability systems are therefore designed to closely mimic the way human analysts reason about trace links and perform tracing tasks [14, 25].

Such systems rely on a knowledge base (ontology) which understands the vocabulary, facts, and assumptions of the domain and represents them in a format that is accessible and processable. Expert solutions must have the ability to reason over the facts of the domain in order to understand when to create a trace link between a source and target artifact and just as importantly when additional constraints and conditions mean that an otherwise plausible link should be rejected. The complexity of most safety critical domains and the expertise needed to reason about trace links in the domain, suggests that expert systems, like humans, will need to be knowledgeable about specific domains.

An expert traceability system also needs to tackle the significant challenge of taking a potentially poorly written artifact, such as a regulatory code or a requirement, removing superfluous information, and automatically mapping its essential content into a highly structured format interpretable by the expert system.

There is currently little work in this area. Zisman et al. developed a rule-based traceability approach which developed a set of rules to generate semantically typed trace links based on parts-of-speech and structural elements of the traced artifacts. Another exception is work by Guo et. al [14] who constructed a preliminary prototype system for tracing in the transportation domain. Their early proof of concept results suggest that if the knowledge base provides sufficient coverage of the concepts found in the source and target artifacts , and if the mapping from the raw text found in those artifacts, to the concepts of the expert system is performed correctly, then there are significant improvements in both recall and precision. The challenge is obviously to create an adequate knowledge base for the domain, a suitable set of link heuristics, and accurate algorithms for performing mappings from raw requirements to the more structured format.

Based on early results, we anticipate that advances in this area will eventually lead to significant breakthroughs in solving the tracing problem. An added advantage is that the expert system can provide human readable rationales for each of the generated links. On the negative side, the cost and effort for achieving this goal is far higher than for delivering tracing solutions at lower levels of the tIQ.

6. KNOWLEDGE SYNTHESIS

The final class of the tIQ represents algorithms which have the ability to synthesize information in order to address the question of *when* each different technique should be used, and *how* various techniques can be combined [11, 9]. Most prior work in synthesis has focused on finding the right blend of various approaches by experimenting with different combinations of term-based techniques. A few recent papers have utilized optimization approaches to configure term-based tracing techniques [27] or to search for the optimal combination of term-based and untyped assocation-based techniques for a given dataset [23].

However, true synthesis of knowledge will only be effective when intelligent traceability solutions from higher levels of our tIQ classification are thrown into the mix. Future systems must be able to go far beyond the basic combinatorial and search-based approaches of today in order to dynamically identify the correct techniques to use to support each individual trace query.

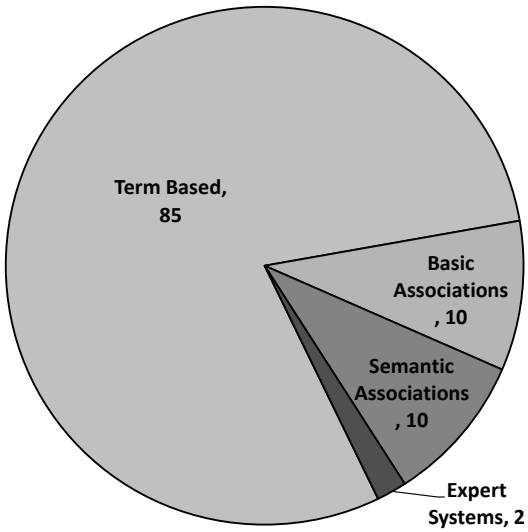

Figure 2: Paper Counts per tIQ category. 4% of papers were also assigned into the cross-cutting synthesis category. These papers were a complete subset of the term-based class.

7. PAST RESEARCH FOCUS

To provide some sense of where research efforts have been targeted in the past ten years, we used *Free Search* to search DBLP for all listed papers from 2004-2013 (inclusive) including the term "Traceability" in either the title, abstract, or list of keywords. DBLP includes the top ranking conferences and journals in which traceability-related work typically occurs, such as the International Conference on Software Engineering (ICSE), the Requirements Engineering Conference (RE), the International Conference on Software Maintenance (ICSM), the Automated Software Engineering Conference (ASE), Foundations of Software Engineering (FSE) or the European Software Engineering Conference (ESEC), IEEE Software, ACM Transactions on Software Engineering Methodology (TOSEM), and IEEE Transactions on Software Engineering (TSE). This search returned 696 results. Any papers related to execution traces or traceability of farm products (or similar) rather than trace retrieval were immediately removed, resulting in a total of 395 papers. We then further narrowed the selection by retaining only those papers which talked about automated trace creation or maintenance. Furthermore, if the underlying trace retrieval algorithm was not clearly explained, typically because the emphasis was on traceability processes or the application of traceability to a specific task, we also eliminated the paper. We were left with a set of 107 papers. Each paper was then classified according to the highest possible level of the tIQ. For example, a paper describing the combination of ontology with term-based approaches was placed into the *Semantically aware* category.

Our study investigated two different issues. First we explored the percentage of research papers that represented each of the tIQ classes. Second, we explored whether there has been an evolution up the tIQ scale as traceability re-

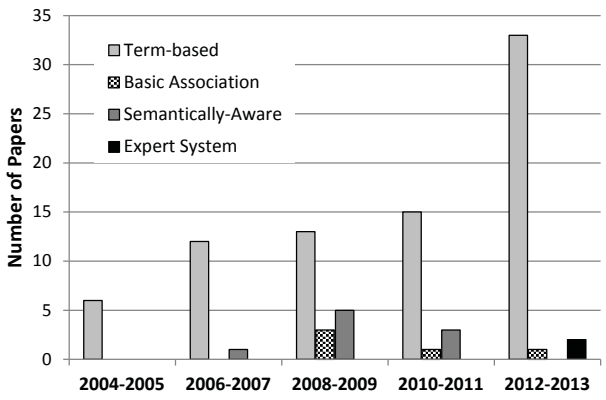

Figure 3: Paper Trends per Category

search has progressed over time.

Figure 2 reports counts of papers categorized into each of the tIQ areas. Out of a total of 107 papers, 81 represented term-based solutions, 10 used basic-associations, 10 used ontology, two implemented expert systems, and 4 used various forms of synthesis.

Figure 3 shows that over the studied period the total counts of papers focusing on trace creation increased from 6 in 2003-2004, to 13 in 2005-2006, 21 in 2007-2008, 19 in 2010-2011, and finally 38 in 2012-2013. While these numbers might have been influenced by overall trends in DBLP demographics, most of the conferences our papers came from appeared in DBLP throughout the entire period of our study. Most research emphasis across all five periods has been on term-based approaches, with a significant and somewhat unexpected peak in the period of 2012-2013. Further analysis showed that this increase was caused by a series of papers which used term-based techniques to create trace links in many different contexts i.e. links to test cases, links in product lines, links in SOA environments. The interest in ontology seems to have peaked around 2007-2008. Papers that explore the use of expert systems are just starting to emerge, with two papers classified this way in the 2012-2013 time period.

However, the most pertinent observation is that the emphasis of the research community in both exploration of new techniques and adoption of existing ones still focuses around term-based approaches. Approximately 76% of papers reporting on trace creation or maintenance techniques describe term-based approaches.

8. NEXT STEPS

In this position paper we have argued that meaningful advances in trace creation algorithms will only be fully realized by integrating more intelligent techniques into the trace retrieval process. Our literature search shows that actual research effort continues to focus on term-based approaches. Given the significant increase in term-based research focus during 2012-2013 we further analyzed those papers and found that 64% of them were adoption related, while only 34% explored new or improved techniques for trace creation and/or maintenance.

It is our belief that continually emphasizing techniques which tweak out small improvements in trace accuracy for certain data sets but not working well for other datasets will not lead to meaningful advances. The exceptions to this 'rule' are techniques which can consistently be applied to repeatedly recognizable *special cases*. For example, algorithms that are able to understand the characteristics of a specific artifact or a specific project context and to apply specific traceability rules and techniques demonstrate levels of intelligence which ultimately may lead to significant advances. Such techniques fall under our tIQ category of *synthesizing knowledge*.

We believe that ongoing work using *term based* approaches will be constrained by the term-mismatch problem and by the inability to reason about complex interdependencies and conditions. This supposition is supported by the fact that over the past 10 years, improvements based purely on such approaches have led to rather small increases in recall and precision, which from the perspective of an industrial tracing problem are not game changers.

From a research perspective, the effort needed to bridge the gap between lower and higher levels of the tIQ is quite extensive. This is not a low-hanging fruit. Expert systems need to be built for non-trivial safety-critical domains governed by regulatory codes and compliance issues. Solutions need to be explored and delivered for interpreting complex software artifacts into the concepts and vocabulary of the underlying knowledge base, so that computerized tracing solutions move closer towards matching the expertise of human analysts. The complexity of the problem suggests the need for highly specialized systems. Such systems might not know everything about avionics, medical devices, or railway communications etc, however they might each understand how a specific set of regulatory guidelines is applied in given domain. Even such a system could be further specialized to work with different types of artifacts such as state diagrams, architectural models, or textual requirements specifications.

A major current limitation to advances in this area is the significant cost and effort needed to build an expert system for even a highly focused area. For example, we are currently developing a proof of concept system for a specific set of regulatory guidelines in an area of the transportation domain [14]. Even with semi-automated tools, the effort to create the system is taking thousands of hours of work. This contrasts with term-based traceability algorithms which can be applied into new domains with minimal effort and almost no customization.

As we increase our ability to use machine learning and information retrieval techniques to construct new ontology from available information sources, we anticipate the use of expert system approaches transitioning from the proof-of-concept stage to more generalized usage. We are far from that point.

We also posit that the final goal is to not only match human expertise, but to surpass it. Studies have shown that the ardor and complexity of the tracing task frequently leads to humans making incorrect tracing decisions [20]. Intelligent tracing solutions are based on rules and facts of the domain which have been carefully scrutinized and agreed upon. They are therefore potentially capable of delivering solutions which surpass those of human trace analysts.

As previously stated, the classification scheme outlined in this position paper is based purely on our subjective analysis of the thought processes and algorithmic solutions that go into automating the tracing process. We therefore welcome further discussion and rigorous debate on this topic.

8.1 Acknowledgments

The work in this paper was partially funded by the US National Science Foundation Grant CCF:1319680.

9. REFERENCES

[1] 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.

[2] N. Assawamekin, T. Sunetnanta, and C. Pluempitiwiriyawej. Ontology-based multiperspective requirements traceability framework. *Knowl. Inf. Syst.*, 25(3):493–522, 2010.

[3] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor. Software traceability with topic modeling. In *ICSE (1)*, pages 95–104, 2010.

[4] B. Berenbach, D. Gruseman, and J. Cleland-Huang. Application of just in time tracing to regulatory codes. In *Proceedings of the Conference on Systems Engineering Research*, 2010.

[5] J. Cleland-Huang, O. Gotel, J. Huffman Hayes, P. Mäder, and A. Zisman. Software traceability: Trends and directions. In *36th International Conference on Software Engineering (ICSE)*, 2014.

[6] 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.

[7] T. Dietrich, J. Cleland-Huang, and Y. Shin. Learning effective query transformations for enhanced requirements trace retrieval. In *ASE*, pages 586–591, 2013.

[8] B. Dit, M. Revelle, and D. Poshyvanyk. Integrating information retrieval, execution and link analysis algorithms to improve feature location in software. *Empirical Software Engineering*, 18(2):277–309, 2013.

[9] D. Falessi, G. Cantone, and G. Canfora. Empirical principles and an industrial case study in retrieving equivalent requirements via natural language processing techniques. *IEEE Trans. Software Eng.*, 39(1):18–44, 2013.

[10] Federal Aviation Authority (FAA). *DO-178B: Software Considerations in Airborne Systems and Equipment Certification*, faa's advisory circular ac20-115b edition.

[11] M. Gethers, R. Oliveto, D. Poshyvanyk, and A. D. Lucia. On integrating orthogonal information retrieval methods to improve traceability recovery. In *ICSM*, pages 133–142, 2011.

[12] M. Gibiec, A. Czauderna, and J. Cleland-Huang. Towards mining replacement queries for hard-to-retrieve traces. In *ASE*, pages 245–254, 2010.

[13] T. Gruber. Ontology. In *Encyclopedia of Database Systems*, pages 1963–1965. 2009.

[14] J. Guo, J. Cleland-Huang, and B. Berenbach. Foundations for an expert system in domain-specific traceability. In *RE*, pages 42–51, 2013.

[15] S. Hayashi, T. Yoshikawa, and M. Saeki. Sentence-to-code traceability recovery with domain ontologies. In J. Han and T. D. Thu, editors, *APSEC*, pages 385–394. IEEE Computer Society, 2010.

[16] 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.

[17] J. H. Hayes, A. Dekhtyar, and S. K. Sundaram. Advancing candidate link generation for requirements tracing: The study of methods. *IEEE Trans. Software Eng.*, 32(1):4–19, 2006.

[18] P. Jackson. *Introduction To Expert Systems (3 ed.)*. Addison Wesley, 1998.

[19] O. G. Jane Cleland-Huang and A. Zisman. *Software and Systems Traceability*. Springer Verlag, 2012.

[20] W.-K. Kong, J. H. Hayes, A. Dekhtyar, and O. Dekhtyar. Process improvement for traceability: A study of human fallibility. In *RE*, pages 31–40, 2012.

[21] N. G. Leveson. *Safeware, System Safety and Computers*. Addison Wesley, 1995.

[22] Y. Li and J. Cleland-Huang. Ontology-based trace retrieval. In *Traceability in Emerging Forms of Software Engineering (TEFSE2013*, San Francisco, USA, May 2009.

[23] S. Lohar, S. Amornborvornwong, A. Zisman, and J. Cleland-Huang. Improving trace accuracy through data-driven configuration and composition of tracing features. In *ESEC/SIGSOFT FSE*, pages 378–388, 2013.

[24] P. Mäder, P. L. Jones, Y. Zhang, and J. Cleland-Huang. Strategic traceability for safety-critical projects. *IEEE Software*, 30(3):58–66, 2013.

[25] A. Mahmoud, N. Niu, and S. Xu. A semantic relatedness approach for traceability link recovery. In *ICPC*, pages 183–192, 2012.

[26] N. Niu and A. Mahmoud. Enhancing candidate link generation for requirements tracing: The cluster hypothesis revisited. In *RE*, pages 81–90, 2012.

[27] A. Panichella, B. Dit, R. Oliveto, M. D. Penta, D. Poshyvanyk, and A. D. Lucia. How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms. In *ICSE*, pages 522–531, 2013.

[28] P. Rempel, P. Mäder, T. Kuschke, and J. Cleland-Huang. Mind the gap: Assessing the conformance of software traceability to relevant guidelines. In *36th International Conference on Software Engineering (ICSE)*, 2014.

[29] Y. Shin and J. Cleland-Huang. A comparative evaluation of two user feedback techniques for requirements trace retrieval. In *SAC*, pages 1069–1074, 2012.

[30] P. Shvaiko and J. Euzenat. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176, 2013.

[31] G. Spanoudakis, A. Zisman, E. Pérez-Miñana, and P. Krause. Rule-based generation of requirements traceability relations. *Journal of Systems and Software*, 72(2):105–127, 2004.