Study of Satisfaction Assessment Techniques for Textual Requirements

K. S. Divya
P.G. Scholar, Department of CSE,
Sri Krishna College of Technology, Coimbatore, India

R. Subha
Assistant Professor, Department of CSE,
Sri Krishna College of Technology, Coimbatore, India

Dr. S. Palaniswami
Principal, Government College of Engineering,
Bodinayakanur, India

ABSTRACT
Requirements satisfaction is an important part in the software development. The right product can only be developed if all the requirements are satisfied. Satisfaction assessment is a process to determine whether all the requirements are satisfied in the design documents. The satisfaction assessment is performed in order to find the satisfied requirements. There are many satisfaction assessment techniques to find the satisfied requirements in the design documents. This paper shows a study of the satisfaction assessment techniques in the textual requirements. A new method is also proposed which implements the semantic diversity to perform the satisfaction assessment. The semantic diversity uses the contextual tracing while performing the candidate satisfaction mapping between the requirements document and design documents.

Keywords
Software Engineering, Requirements Engineering, Requirements tracing, Satisfaction assessment, Term frequency, Parts of speech tagging

1. INTRODUCTION
Software engineering is the application of a systematic and disciplined approach to the design, development, operation, and maintenance of the software. Requirements engineering is a sub discipline of software engineering.

1.1 Requirements Engineering
Requirements are the statements about the system activities, system behavior, system properties, system qualities and the system constraints. Requirements engineering is the study under the software engineering. Requirements engineering defines the use of systematic and
disciplined techniques that ensures that the requirement is complete, consistent and correct. Requirements engineering consist of many activities such as requirements elicitation, analysis, specification, verification, and management, where:

- **Requirements elicitation** is the process of determining and understanding the needs of the customers.
- **Requirements analysis** is the process of checking the customer needs.
- **Requirements specification** is the process of representing the customer needs in the document format.
- **Requirements verification** is the process of checking that the system requirements are complete, correct, consistent, and clear.
- **Requirements management** is the process of scheduling, coordinating, and documenting all the requirements engineering activities (that is, elicitation, analysis, specification, and verification).

### 1.2 Problems in Requirements Engineering

Requirements engineering is recognized as a critical task, since many software failures originate from inconsistent, incomplete or simply incorrect requirements specifications. Many of the common, most serious problems associated with software development are related to requirement. The main problem occurs in the requirements elicitation process. The problems are:

- **Problems of scope**
  These problems occur when there is too little information or too much information. Sometimes unnecessary design information may also be given in the requirements document.
- **Problems of understanding**
  These problems occur when users have incomplete understanding about their needs and conflicts views between others.
- **Problems of volatility**
  These problems occur when the requirements change due to change in time.

The problems can also arise from requirements specification and requirements validation and verification.

### 1.3 Requirements Traceability

Requirements traceability is a sub-discipline of requirements management within software development and systems engineering. Requirements traceability is concerned with documenting the requirements and providing the bi-directional traceability between various associated requirements. It enables users to find the origin of each requirement and track every change that was made to this requirement.
1.4 Satisfaction Assessment

Satisfaction assessment is defined as the process of performing the satisfaction mapping of portions of textual requirements to design elements which is represented in natural language. A satisfaction mapping contains a satisfaction decision that has been made about a set of textual requirement elements and a set of corresponding textual design elements. Satisfaction assessment determines whether every component in the design document is addressed by the component in the requirement document.

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>identifying each requirement</td>
</tr>
<tr>
<td>2</td>
<td>assigning a unique identifier to each requirement</td>
</tr>
<tr>
<td>3</td>
<td>for each high level requirement, determine all matching low level requirements</td>
</tr>
<tr>
<td>4</td>
<td>for each low level requirement, determine a parent requirement in the high level document</td>
</tr>
<tr>
<td>5</td>
<td>determine if each high level requirement have been completely satisfied</td>
</tr>
</tbody>
</table>

Formal Definition

Given a set of requirements decomposed into terms \( R = \{r_1, r_2, \ldots\} \) and a set of design element terms \( D = \{d_1, d_2, \ldots\} \). A satisfaction mapping is a set of pairs of terms \( (r_m, d_m) \) where \( r_m \) is a term in a set of requirements and \( d_m \) is a term in the set of design elements where \( r_m \) is directly correlated to \( d_m \).

This paper proceeds as follows. In the next section, the related works in this field are described. Section 3 describes satisfaction assessment techniques. Section 4 describes the proposed system. Section 5 provides the conclusion.

2. RELATED WORKS

Elizabeth Ashlee Holbrook, Jane Huffman Hayes, Alex Dekhtyar, Wenbin Li [1] explains various methods for the satisfaction assessment in the textual requirements. Satisfaction assessment helps in identifying the unsatisfied requirements. First the requirement traceability matrix for the data set is constructed and the requirement and design text is converted into chunks. Stop word removal and the stemming for the chunks are performed. The chunks are tokenized into individual terms. The synonym pairs for the terms
are determined. For TF-IDF and Naive Satisfaction method, the threshold values are predefined. For NLP satisfaction method, the rules are generated. Finally the candidate satisfaction assessment mapping is performed to determine the satisfied candidates.

Holbrook E A, Hayes J H, Dekhtyar A [2] explains the automatic methods for satisfaction assessment. The system introduces the automation of satisfaction assessment which is the process of performing the satisfaction mapping of textual requirements to design elements which is represented in natural language. The system describes the satisfaction assessment approach algorithmically and then evaluates the effectiveness of two proposed information retrieval (IR) methods in two industrial studies. Mainly focuses on assessing whether requirements have been satisfied by lower level artifacts such as design.

Hayes J H, Dekhtyar A, Sundaram S, Holbrook A, Vadlamudi S [3] explains a tool for the requirements tracing. The recovery of traceability for artifacts containing unstructured textual narrative is addressed. RETRO uses information retrieval (IR) and text mining methods to construct candidate traces. The task is to find documents in the collection that are deemed relevant to the query. The method vector space retrieval with tf-idf term weighting is the default tracing technique in RETRO. Stop word removal is performed for every document and query.

Jane Huffman Hayes, Alex Dekhtyar, Senthil Karthikeyan Sundaram [4] explains the candidate link generation for requirements tracing. The goals for a tracing tool based on analyst responsibilities in the tracing process are defined. The several new measures for validating that the goals have been satisfied are introduced. The analyst feedback in the tracing process is implemented. A prototype tool, RETRO (REquirements TRacing On-target), to address the goals is presented. The methods and tool can be used to trace any textual artifact to any other textual artifact. An additional IR technique, Latent Semantic Indexing is used for requirements tracing. A requirements tracing tool is defined that is special purpose software that takes as input two or more documents in the project document hierarchy and outputs a traceability matrix that is a mapping between the requirements of the input documents. Two IR algorithms TF-IDF vector retrieval and vector retrieval with a simple thesaurus one newly implemented method, Latent Semantic Indexing are used for determining requirement similarity. LSI is a dimensionality reduction method, which allows one to capture the similarity of underlying concepts, rather than simple keyword matches.
Robinson W N [5] explains the implementation of rule based monitors for requirements tracing. A language for requirements and monitor definitions are defined by the framework. A methodology for defining requirements, identifying potential requirements obstacles, and analyzing monitor feedback is defined. The framework address three interrelated monitoring issues such as Formalization of high-level goals, requirements, and their monitors, Automation of monitor generation, deployment, and optimization and Traceability between high-level descriptions and lower-level run-time events. The monitoring approach integrates requirements language research with commercial business process monitoring. The approach defines the logical monitoring model. The goals and requirements are defined. Potential requirements obstacles are uncovered and their monitors are derived. The monitoring architecture and implementation are defined. The requirements of the monitoring event sources and sinks are defined. A logical-physical mapping to ensure traceability of events back to requirements is defined. The monitoring system is implemented and deployed. The high-level feedback on the systems actions and requirements compliance is provided. The compensation and adaptation rules are executed when violations occur. The high-level feedback on the monitoring system itself, thereby providing historical information used in defining new monitoring optimization rules is provided.

Marcus A, Maletic J I [6] explains the latent semantic indexing method. A method to recover traceability links between documentation and source code, using an information retrieval method, namely Latent Semantic Indexing (LSI) is presented. The traceability links based on similarity measures are identified. The method utilizes all the comments and identifier names within the source code to produce semantic meaning with respect to the entire input document space. The vector space model (VSM) is a widely used classic method for constructing vector representations for documents. Latent Semantic Indexing (LSI) is a VSM based method for inducing and representing aspects of the meanings of words and passages reflective in their usage. LSI uses a user constructed corpus to create a term-by-document matrix. New document vectors (and query vectors) are obtained by orthogonally projecting the corresponding vectors in a VSM space (spanned by terms) onto the LSI subspace. The LSI subspace captures the most significant factors (i.e., those associated with the largest singular values) of a term-by-document matrix, it is expected to capture the relations of the most frequently co-occurring terms.

effectively is described. A method for establishing and utilizing traceability links between requirements and performance models is proposed. Traceability links are established through the use of a dynamic traceability scheme capable of speculatively driving the impacted models whenever a quantitative requirement is changed. Key values from within the individual performance models representing probabilities, rates, counts and sizes etc are placed in the central requirements repository. Finely tuned links are then established between the data-values in the models and those in the repository. The process of analyzing the impact of a proposed change upon the performance of the system through dynamic re-execution of requirement dependent models is supported.

Giuliano Antoniol, Gerardo Canfora, Gerardo Casazza, Andrea De Lucia, Ettore Merlo [8] explains the traceability between code and documentation. A method based on information retrieval to recover traceability links between source code and free text documents is proposed. The method proposed ranks the free-text documents against queries constructed from the identifiers of source code components and can be customized to work with different IR models. Both a probabilistic and a vector space information retrieval model are applied. In the probabilistic model, free-text documents are ranked according to the probability of being relevant to a query computed on a statistical basis. A language model for each document or identifiable section is estimated and uses a Bayesian classifier to score the sequences of mnemonics extracted from each source code component against the models. The vector space model treats documents and queries as vectors in an n-dimensional space. Documents are ranked against queries by computing a distance function between the corresponding vectors. The documents are ranked according to a widely used distance function, i.e., the cosine of the angle between the vectors. The construction of the vocabulary and the indexing of the documents are preceded by a text normalization phase performed in three steps. In the first step, all capital letters are transformed into lower case letters. In the second step, stop-words (such as articles, punctuation, numbers, etc.) are removed. In the third step, a morphological analysis is used to convert plurals into singulars and to transform conjugated forms of verbs into infinitives. The construction of a query consists of three steps. Identifier extraction parses the source code component and extracts the list of its identifiers. Identifier separation splits identifiers composed of two or more words into separate words. Text normalization applies the three steps described above for document indexing. Finally, a classifier computes the similarity between queries and documents and returns a ranked list of documents for each source code component.
3. SATISFACTION ASSESSMENT TECHNIQUES

3.1 Naive Satisfaction Assessment

The naive satisfaction method is based on a simple idea of tracking and thresholding the percentage of common terms between the two chunks. This method is simple and easy to implement. The naive satisfaction method determines the root of the elements in the requirements document and the design document. If the terms in the requirement chunk and the design chunk contain the same root or the root of the synonym then the terms are considered as a match pair. The similarity value for the element pairs is determined.

\[ \text{sim value} = \frac{\text{Number of times the term occur}}{\text{total number of terms in the document}} \]  

Threshold values from 0.01 to 0.09 are used to filter the chunks. The chunks with similarity values below the threshold values are excluded from the candidate satisfaction assessment mapping.

Drawbacks

The naive satisfaction approach only determines the textual similarity. The polysemous words are also treated as the textual similar words.

3.2 TF-IDF Satisfaction Assessment

The TF-IDF satisfaction assessment method is based on vector space information retrieval using TF-IDF (term frequency - inverse document frequency) term weighting. TF-IDF method is the traditional information retrieval method which is commonly used in requirements tracing. TF-IDF is the measure of the importance of a term within a document. Term frequency (TF) is the number of times a particular term appears within a document. Inverse document frequency (IDF) of a term is the logarithm of the ratio of the total number of documents in a collection to the number of documents that contain the term. Each requirement and design element chunk is considered an individual document within the document collection. TF-IDF similarity scores are calculated between pairs of requirement chunks and design chunks.

\[ \text{IDF} = \log \left( \frac{N}{DF} \right) \]  
\[ \text{TF-IDF} = \text{TF} \times \text{IDF} \]

Threshold values from 0.01 to 0.09 and 0.1 to 0.9 are used to filter the chunks. The chunks with similarity score below the threshold values are excluded from the candidate satisfaction assessment mapping.
Drawbacks
The TF-IDF satisfaction approach only determines the similarity based on the importance of the terms within the document. The polysemous words are also treated as similar words in the TF-IDF approach.

3.3 NLP Rule Based Satisfaction Assessment

NLP rule based satisfaction assessment is based on certain rules that are defined by the user. For the NLP rule based satisfaction assessment the parts of speech for all the elements are determined using parts of speech tagging. Parts of speech represent the structure of the sentences. Rules are created to help in identifying the requirement element and design element matching pair. The rule set can be created manually using the text editor. A rule is specified in the following format:

[Element1Position][Element1PartofSpeech][Element1Type][Element2Position][Element2PartofSpeech][Element2Type][MinSimilarity][Confidence][Enabled]

For example, the rule:
Any|NP|RE|First|VP|DE|45|20|True

specifies that if any noun phrase in a requirement chunk is at least a 45% match based on lexical similarity with the first verb phrase in a design element, then the requirement chunk and design element chunk should be paired with 20% confidence.

Drawbacks
The NLP rule based approach determines the similarity based on their structure. The polysemous words are also treated as similar words in the NLP rule based approach.

3.4 Candidate Link Generation Method

Requirement tracing first begins with the parsing of documents. Candidate link generation is performed to determine the matching pair of requirement elements and the design elements. Then the candidate link evaluation is performed to evaluate the measurement of candidate link lists. To determine the candidate mapping, first the elements are extracted from the requirements document as well as from the design document. Then keywords are assigned to each requirement document and each design document. The keyword assignment can be performed either manually or
using the search functions from the word processor or spreadsheets. Then the candidate links are determined. To determine whether the design element is matching with the requirement element is performed using a key-word matching algorithm. Recall and precision are the two measures used to perform the candidate link evaluation. The candidate link evaluation determines whether the candidate links are true links or false links.

Drawbacks
The candidate link generation approach determines the similar words using the key word matching algorithm. The polysemous words are also treated as similar words in the candidate link generation method approach.

3.5 RETRO Tool

Requirements tracing on-target is a special purpose requirement tracing tool. RETRO uses information retrieval (IR) and text mining methods to construct candidate traces. RETRO. The core part of RETRO consists of the IR toolbox, the feedback processing methods, and the GUI front end. The methods for building representations of traced documents are also included. There are two modes for tracing the requirements in RETRO. In automatic tracing mode, the candidate links are generated using automated methods. The manual tracing mode provides the ability to browse high- and low-level documents for the purpose of discovery of any links not found by the automated tools. The filtering tool allows reducing the display of the candidate link lists. A threshold value would be specified by the analyst.

The low level documents that have the weight value greater than the threshold value would be displayed. The threshold is controlled by a slider bar that can be moved between 0 and 1 in the intervals 0.01. The filter that is selected could have global effect or local effect. When the filter is in global effect, the current filter value is applied to candidate link lists belong to all high-level elements. When the filter is in local effect, the current filter value is applied only to the candidate link list belongs to the currently selected high-level element. The candidate links could be displayed in three ways. First, the links could be displayed one at a time. Second, all the candidate links could be displayed in the order they appear in the design document. Third, the candidate links could be displayed in their relevance order.

Drawbacks
The RETRO approach determines the similar words using the information retrieval and text mining method. The polysemous words are also treated as similar words in the RETRO tool approach.
4. PROPOSED WORK
The existing satisfaction assessment techniques such as Naive satisfaction assessment, TF-IDF satisfaction assessment, NLP Rule based satisfaction assessment, Candidate link generation and RETRO tool performs the candidate satisfaction mapping. These methods only determine the satisfied candidates using the similar words. The semantic similarity is not determined using the existing systems. The proposed system implements the concept of semantic diversity to determine the satisfied candidates. To determine the semantic similarity, latent semantic analysis is performed to the requirements document and the design documents. Then the semantic diversity for the elements is calculated. Semantic diversity is the degree to which the different contexts associated with a given word vary in their meanings.

5. CONCLUSION
In the software development, requirements satisfaction plays an important role. All the requirements specified by the users should be satisfied. Satisfaction assessment determines whether all the requirements are addressed in the design document. There are many methods to perform the satisfaction assessment to identify the satisfied requirements. Naive satisfaction assessment method is based on the textual similarity. TF-IDF satisfaction assessment is based on the importance of a term in a document. NLP Rule based satisfaction assessment is based on certain user defined rules. Candidate generation links generates the satisfied candidates based on the information retrieval and the text mining method. The RETRO tool is used to perform automated requirements tracing. The existing system does not perform any contextual tracing. The semantic diversity could be implemented to perform the contextual tracing to determine the satisfied candidates.

REFERENCES


This paper may be cited as: