

On Presuppositions in Requirements

Lin Ma Bashar Nuseibeh Paul Piwek Anne De Roeck Alistair Willis

*Department of Computing, The Open University, U.K.
{L.Ma, B.Nuseibeh, P.Piwek, A.Deroeck, A.G.Willis} @open.ac.uk*

Abstract

Tacit knowledge in requirements documents can lead to miscommunication between software engineers and other stakeholders. One way in which the presence of tacit knowledge is signalled in text is by linguistic presuppositions. In this paper, we present a brief introduction to tacit knowledge, presuppositions and the links between them. Our aim is to build a theoretically grounded system which is able to automatically highlight all the presuppositions that might have a negative impact on communication through requirements documents.

1. Introduction

Tacit knowledge can be seen as knowledge not yet put into words. Similarly, the presuppositions of a sentence are assumptions that the author or speaker takes for granted and they are signalled by specific types of syntactic structure. In requirements documents, tacit knowledge can lead to miscommunication between software engineers and other stakeholders. One way in which the presence of tacit knowledge can be discovered in text is through presuppositions. For instance, a requirement “Radiation dose expected is less than 1 krad and 10^{10} n/cm²” commits the reader to the radiation dose being expressible in krad and n/cm². Tracking such implicit commitments in requirements documents, and identifying which of those are not made explicit, can reveal elements of tacit knowledge.

In this paper, we describe the results of our preliminary investigation into observing and analyzing the presence, characteristics and potential impact of presuppositions. We propose to build a systematic method for making presuppositions explicit in requirements documents in order to avoid misunderstanding and misinterpretation when communicating requirements. The paper provides a short review of tacit knowledge and presuppositions, and then summarizes our observation of presuppositions in requirements.

2. Tacit Knowledge

Polanyi characterizes tacit knowledge as “knowing more than one can tell” [1]. This way, however, tacit knowledge is defined purely and exclusively in terms of “tacitness”, which means, in contrast with explicit knowledge, that tacit knowledge cannot be articulated at all, and therefore cannot be used or managed by human beings [2]. Gourley [3] credits Janik [4] with the proposal that the term “tacit knowledge” can be used at two different levels: firstly, following the Polanyi’s view, tacit knowledge is knowledge inexpressible in words, and acquired by familiarity or practice (such as smells and sounds). Secondly, tacit knowledge at a shallow level is knowledge not yet put into words such as craft knowledge and *presuppositions*. Although there are other ways to define dimensions and hierarchies of tacit knowledge, such as the cognitive and technical dimensions [5], we adopt Janik’s second perspective as our working definition. In our case, tacit knowledge is knowledge that stakeholders know and could have articulated but have not for some reason, perhaps because they simply were not asked. The portion of tacit knowledge that we are focusing on resides between the completely tacit level and the explicit level; in other words, the codifiable and manageable part of tacit knowledge.

In requirement engineering, when key pieces of tacit knowledge are not apparent in requirements documents, it can make communication difficult. This can be particularly problematic when software engineers focus on the development of software, while important details and assumptions in other stakeholders’ domains are overlooked.

3. Presupposition

One way of determining that tacit knowledge is present is by looking at mechanisms in text for making oblique references to bits of knowledge that are taken for granted. One such mechanism is presupposition. In this section, we introduce the notion of presupposition and its potential impacts on communication. Before we describe

presupposition, let us first introduce the notion of “context” that we use in this article. Bunt [6] classifies different types of context into five general categories: the linguistic context, the semantic context, the physical context, the social context and the cognitive context. In this paper, for presupposition, we employ two of them: the linguistic context which consists of the preceding sentences in a discourse, and cognitive context including beliefs, speakers’ attitudes and world knowledge.

Presupposition is a linguistic phenomenon that dominates the largest portion of literature in linguistic pragmatics [7]. It has been a central topic in both semantics and pragmatics, since Frege was first dealing with such problems. When someone produces an utterance¹ - for example, an assertion, question or command - there can be accompanying assumptions that the speaker or author presupposes, that is, which are taken for granted. The following are classical and representative examples of assertions studied by presupposition theorists:

- (1) a. *The King of France is bald.*
- b. *Richard managed to pass the exam.*
- c. *Mark has stopped beating his wife.*

When a speaker utters these sentences, he/she naturally presupposes that there is a King of France, Richard tried to pass the exam and Mark used to beat his wife respectively².

The study of presupposition is in fact the study of not only natural language but also the interaction between people. In other words, speakers rather than statements presuppose [9, 10]. In particular, Stalnaker [9] defines the notion of pragmatic presupposition as follows: “A speaker presupposes that P at a given moment in a conversation just in case he is disposed to act, in his linguistic behaviour, as if he takes the truth of P for granted, and as if he assumes that his audience recognizes that he is doing so.” In his definition, it is the speaker who presupposes something; the sentence itself is just a means of implicitly expressing the speaker’s presupposition.

There appear to be two major problems³ in the study of

¹ An utterance is the form of realizing language in one’s individual concrete way [8]. Here we simply use this notion to represent a unit of speech or text under study.

² There are other presuppositions in these sentences, but the focus here is on different types of presupposition.

³ In fact, there is a third major problem, called presupposition failure [14], which is how to judge the truth value of an

presupposition: presupposition projection and presupposition accommodation. An elementary presupposition is a presupposition of part of an utterance. Presupposition projection, as the name suggests, is the study of whether an elementary presupposition is a presupposition of the whole utterance. Here two examples are given for distinct scenarios in requirements, one where an elementary presupposition projects out and one where it does not:

- (2) a. *If funds are inadequate, the system will notify....*
- b. *If there is a system, the system will notify...*

Intuitively, when a hearer accepts utterance (2b), he/she does not take the presupposition that there is a system for granted. The elementary presupposition that there is a system in the consequent of the conditional somehow does not project. The same elementary presupposition that there is a system nevertheless projects out in example (2a), which signals to the hearer that the speaker takes for granted that there is a system. These examples show that presuppositions sometimes do and sometimes do not project out. The projection problem is the problem of providing a systematic account of when presuppositions do and when they do not project out. The tendency of presuppositions to project out when embedded in a complex sentence is a distinctive property of presuppositions. Although other linguistic phenomena may sometimes project out as well, - for example conversational implicature - presupposition is the only one that normally tends to project when embedded in various complex sentences containing negation, modal operators, etc. [10, 11]. This property has proved to be a useful tool for testing whether certain types of syntactic structure generate presuppositions, that is, are *presupposition triggers*. These types of syntactic structure include, for example, the *definite description* “the King of France” in example (1a) and the *implicative verb* “manage” in example (1b).

Van der Sandt’s Binding Theory [12] provides an effective, though not perfect⁴, explanation of the projection problem. Binding Theory makes use of discourse representation theory [13] and treats, presuppositions as anaphora (an expression, such as a pronoun, which depends for its interpretation on a preceding expression, i.e., an antecedent). Presupposition

utterance when a presupposition of the utterance is false. This is still debated and beyond the scope of this paper.

⁴ According to Spender’s [15] corpus study, the Binding Theory lacks an explanation to presuppositions triggered by the particle ‘too’. And it also lacks the consideration of world knowledge [17].

projection is treated as looking for a path to an earlier part of the discourse which hosts an antecedent that can bind the presupposition⁵. Whenever an antecedent is found in the discourse, the presupposition is bound, and thus does not project out. For the presupposition that there is a system, the presupposition triggered by “the system” in example (2b), an antecedent is found in the clause “If there is a system”. However, for example (2a), no antecedent can be found in either the sentence or the linguistic context (which we assume to be empty). This brings us to the second problem of presupposition – presupposition accommodation.

According to the Binding Theory, accommodation makes presupposition triggers slightly different and more powerful in expression than pronominal anaphora. For a single pronoun such as “he”, if there is no antecedent explicitly provided in the discourse, it cannot be resolved. Unlike pronouns, a presupposition has its own descriptive content, so that it has the ability to let a hearer spontaneously generate an antecedent to be accommodated into the cognitive context in order for an unbound presupposition to be bound with. For example, in this requirement [16]:

(3) *The ROW chief initiates the database record.*

The presupposition that there is a database record triggered by the definite description “the database record” cannot be bound with any explicit antecedent. The hearer has to mentally construct “a database record” in his individual cognitive context to enable the binding process for following the discourse. However, we note that if there is a relevant antecedent existing in the hearer’s cognitive context, the hearer may wrongly choose to bind rather than to accommodate. More precisely, the reader of this requirement may know a database record *A* and choose to believe *A* is the record that the document author is writing about. However, the document author may mean database record *B* or just a new database record. Without further communication, they can arrive at different interpretations, each believing they are agreed on a same database record.

According to Stalnaker’s pragmatic definition of presupposition, presupposition can point to underlying information that is taken for granted and hence is tacit. Taking examples from a requirements document [16], with the linguistic context set empty, which means these sentences appear at the beginning of the document:

⁵ This is analogous to the notion of “anaphora resolution” in linguistics [12].

- (4) *a. The RPC readout shall be sent to the terminal.*
b. Accessibility in the experimental hall is required for changing the piggy board where the device will be mounted.

In example (4a) the document writer presupposes that there is an RPC readout and that there is a terminal. The absence of either “an RPC” or “a terminal” is likely to cause readers to arrive at different interpretations. In addition, if the acronym, RPC, as a regular shorthand expression used daily by engineers, is not defined in the document explicitly, there could be difficulties in understanding for readers. Moreover, as Piwek & Krahmer [17] point out, world knowledge is often essential for resolving presuppositions. In example (4b), the document reader must have the knowledge that the piggy board is a kind of device in order to understand this requirement correctly. Different intentions - examples (3) and (4a) - and different world knowledge - example (4b) - between stakeholders and the document writer may possibly lead to different interpretations of a piece of presupposed information without being noticed. Therefore, we propose that a large amount of tacit knowledge can be extracted by detecting and analyzing presuppositions existing in requirements, and that highlighting presuppositions may allow us to mitigate the negative effects of tacit knowledge.

4. Presuppositions in Requirements

In computational linguistics, presupposition is believed to be signalled by certain types of syntactical structure. These are called presupposition triggers [7]. In our preliminary work so far, we examined one 20-page requirements document [16] to determine the presence or absence of each kind of presupposition trigger. The trigger types we used to identify presuppositions in this document were collected from a list of some well-known and representative presupposition triggers [7]. Table 1 shows the result of the study of this document and the trigger types used.

In this document, presuppositions triggered by definite descriptions appeared 399 times. This highlights the volume of presupposed information in requirements documents. We expected to find examples of all the trigger types in the list we used. However, apart from definite descriptions, some of the trigger types were absent or few in this document. The absent trigger types include: change of stage verb (e.g. continue, stop), cleft (It + be + noun + subordinate clause), stressed constituents (words in italic in texts) and counterfactual conditionals (what would be the case if something were true). As a preliminary validation of this distribution of presuppositions, another requirements document was

checked, which suggests that the results are consistent.

Some example sentences (S) are taken from the document coupled with their presuppositions (P) including (trigger in bold):

(5) S: ...tests **revealed** that redundancy to Single Event Upsets is required. (factive verb)

P: redundancy to Single Event Upsets is required.

S: ...chambers shall **avoid** that two CMA share the same gas volume...(implicative verb)

P: two CMA may share the same gas volume.

Trigger Types	Examples of Triggers from the Document	Number of Occurrences
Definite Description	“the terminal”	399
Factive verb	“reveal”	1
Implicative verb	“avoid”	3
Expression of repetition	“also”	1
Temporal relation	“since”, “after”	2
Comparison	“less/larger than”	3
Question	questions presenting alternatives	1

Table 1. Different types of triggered presuppositions, in the requirements document

We used the Stanford Lexicalized Parser [18] to help track definite descriptions. The documents were parsed into grammatical parse trees, so that definite descriptions starting with determiners like “the”, “that” and “his” etc. could easily be found automatically. Unfortunately, most other kinds of presupposition triggers can only be discovered by hand at the moment, because there is no full vocabulary of presupposition triggers. Currently, we have defined the categories in terms of a few representative example words or constructions. In our future work, for these undocumented trigger types, we will be looking for an automated way to detect them based on natural language processing techniques.

One example that is of particular interest is mentioned in section 3. It concerns an acronym, “the RPC”, which is not defined within the document. The existence of an RPC is, however, presupposed as a result of the presence of the definite noun phrase “the RPC”; in other words, “the RPC” is a presupposition trigger. For a reader, an undefined acronym is to some extent similar to an unknown noun. There is, however, more scope for

misunderstandings or confusion when an undefined acronym is involved. Whereas noun meanings are usually closed - there is typically only one or very few interpretations - acronyms often have many different meanings. Consequently, there is a higher risk that the reader will pick out an unintended meaning. For example, the acronym ATM has at least 25 meanings according to Wikipedia [19] from domains including science, technology, organizations and health. Within the technology domain, there are definitions such as: Automated Teller Machine, Asynchronous Transfer Mode, Alternating Turing Machine, Apollo Telescope Mount and Adobe Type Manager. We tested our hypothesis regarding the difference between unknown nouns and undefined acronyms informally by asking three persons with no technological domain knowledge to read the following sentences:

(5) a. *The peptide bond has two resonance forms*

b. *The ATM was one of a number of projects that came out of the late 1960's.*

For the first sentence (5a), two of the participants were curious about the meaning of “peptide bond”. But for the second sentence (5b), all of respondents took the definition of ATM that they know the best for granted, which is Automated Teller Machine. The sentence was, however, extracted from a Wikipedia article [20] describing the Apollo Telescope Mount. This suggests, unsurprisingly, that there are circumstances in which undefined acronyms can lead to misinterpretation by some readers. What this also suggests is that when we consider presuppositions as a cause for misunderstandings and confusion, we should examine not just the interpretation of the presupposition trigger as a whole, but also its component parts.

5. Future work

We plan to focus on developing an automatic approach for tracing presupposition triggers so as to reveal tacit knowledge in requirements. So far, we have collected some elementary presuppositions in requirements documents. After finding a way of tracing presupposition triggers, we need to investigate building binding relations between those elementary presuppositions. With a collection of presuppositions in requirements and an automatic tool, software engineers and domain experts can be asked to judge if each type of triggered presupposition that contains tacit knowledge impacts negatively on the quality of requirements documents.

In the MaTREx⁶ project, we will use presupposition analysis with ambiguity analysis in requirements documents to detect the possible existence of tacit knowledge. These two kinds of analysis, both of which have to do with management of requirements information, are related to each other, since ambiguously presupposed information could result in misinterpreted assumptions. According to the Binding Theory, when no specific binding can be established because a number of candidates are equally plausible, the text remains uninterpreted. For instance:

- (6) a. A communication requirement and a performance requirement will be discussed in this document.
 b. The requirement is difficult to express.

In this example it is unclear whether “the requirement” refers to the communication or the performance requirement. Subsequently, more documents will be analyzed and presuppositions will be collected to observe their behaviour and linguistic attributes for building a system. With the help of natural language processing techniques, our aim for this system is to help both readers and writers track presuppositions so as to mitigate the negative effects of tacit knowledge.

6. Conclusion

Our preliminary work has shown that various presuppositions exist in requirements documents and some may be unclear to readers. We highlighted several cases and one of which involved undefined acronyms. Future work will focus on effective ways of automatically finding presuppositions in requirements, and the potential existence of tacit knowledge behind them. In particular, we aim to build a system that can automatically highlight presuppositions which have a negative impact on communication in requirements documents.

References

- [1] M. Polanyi, *The Tacit Dimension*, New York, 1967.
- [2] A. Stone and P. Sawyer, “Identifying Tacit Knowledge-Based Requirements”, *IEE Proceedings Software*, vol. 153, 2006, p. 211.
- [3] S. Gourlay, “Tacit Knowledge, Tacit Knowing or Behaving?”, *Proceedings of The Third European Conference on Organizational Knowledge, Learning and Capabilities (OKLC 2002)*, 2002, pp. 5-6.
- [4] A. Janik, “Tacit Knowledge, Working life and Scientific Method”, *Springer Series On Foundations And Applications of Artificial Intelligence*, 1988, pp. 53-63.
- [5] I. Nonaka and H. Takeuchi, *The Knowledge-creating*

Company, New York, vol. 1, 1995, p. 995.

- [6] H. Bunt, “Dynamic Interpretation and Dialogue Theory”, *The Structure of Multimodal Dialogue*, vol. 2, 1995, pp. 139-166.
- [7] S.C. Levinson, *Pragmatics*, Cambridge University Press, 2000.
- [8] M.M. Bakhtin, “The Problem of Speech Genres”, *Speech Genres and Other Late Essays*, 1986, pp. 60-102.
- [9] R. Stalnaker, “Presuppositions”, *Journal of Philosophical Logic*, vol. 2, 1973, pp. 447-457.
- [10] B. Geurts, “Presupposition”, *Presuppositions and Pronouns*, Elsevier Science, 1999, pp. 1-30.
- [11] D.I. Beaver, *Presupposition and Assertion in Dynamic Semantics*, CSLI Publications, 2001.
- [12] R.A. van der Sandt, “Presupposition Projection as Anaphora Resolution”, *Journal of Semantics*, vol. 9, 1992, pp. 333-377.
- [13] H. Kamp, “A Theory of Truth and Discourse Representation”, *Formal Methods in The Study of Language*, 1981, pp. 277-322.
- [14] R. Garner, “The Theory of Presupposition Failure”, *Language*, vol. 54, 1978, pp. 973-977.
- [15] J. Spenader, “Presuppositions in Spoken Discourse”, Phd. Thesis, Department of Linguistics Stockholm University, 2002.
- [16] Barrel LVL1 Muon Trigger Coincidence Matrix ASIC: User Requirement Document. Available: <http://sunset.roma1.infn.it/muon1/docs/notes/com-da-2000-050.pdf>. Accessed: June, 2009
- [17] P. Piwek and E. Kraemer, “Presuppositions in Context: Constructing Bridges”, *Formal Aspects of Context*, vol. 20, Kluwer Academic Publishers, 2000.
- [18] M.C.D. Marneffe, B. MacCartney, and C.D. Manning, “Generating Typed Dependency Parses From Phrase Structure Parses”, LREC 2006, 2006.
- [19] “ATM”, Wikipedia, the free encyclopedia. June 2009. [online]. Available: <http://en.wikipedia.org/wiki/ATM>
- [20] “Apollo Telescope Mount”, Wikipedia, the free encyclopedia, Feb. 2002. [online]. Available: http://en.wikipedia.org/wiki/Apollo_Telescope_Mount

⁶ EPSRC research grant EP/F068859/1