Software Qual J (2011) 19:461–486
DOI 10.1007/s11219-010-9123-6

SQM 2009

The role of comprehension in requirements and implications for use case descriptions

Keith Phalp · Anita Adlem · Sheridan Jeary · Jonathan Vincent · John Kanyaru

Published online: 7 December 2010

Abstract Within requirements engineering, it is generally accepted that in writing specifications (or indeed any requirements phase document), one attempts to produce an artefact which will be simple to comprehend for the user. That is, whether the document is intended for customers to validate requirements, or engineers to understand what the design must deliver, comprehension is an important goal for the author. Indeed, advice on producing ‘readable’ or ‘understandable’ documents is often included in courses on requirements engineering. However, few researchers, particularly within the software engineering domain, have attempted either to define or to understand the nature of comprehension and its implications for guidance on the production of quality requirements. Therefore, this paper examines thoroughly the nature of textual comprehension, drawing heavily from research in discourse process, and suggests some implications for requirements (and other) software documentation. In essence, we find that the guidance on writing requirements, often prevalent within software engineering, may be based upon assumptions that are an oversimplification of the nature of comprehension. Hence, the paper examines guidelines which have been proposed, in this case for use case descriptions, and the extent to which they agree with discourse process theory, before suggesting refinements to the guidelines which attempt to utilise lessons learned from our richer understanding of the underlying discourse process theory. For example, we suggest subtly different sets of writing guidelines for the different tasks of requirements, specification and design.

Keywords Requirements · Comprehension · Discourse process · Use case · Specification · UML

K. Phalp (✉) · A. Adlem · S. Jeary · J. Vincent · J. Kanyaru
Software Systems Research Centre, Bournemouth University, Fern Barrow,
Poole, Dorset BH12 5BB, UK
e-mail: kphalp@bournemouth.ac.uk
URL: http://www.bournemouth.ac.uk/ssrc/

1 Introduction: the importance of comprehension

The complete, accurate and concise documenting of requirements is of vital, perhaps paramount importance (Glass 1998) within software development, and errors made in this phase are often considered the most difficult to solve and most costly to fix (Bray 2002). Hence, the potential benefits of successful comprehension (Graesser et al. 1994) promise improvements in software quality, stakeholder satisfaction and development costs. However, comprehension is 'a complex interaction of basic cognitive processes' (Fletcher et al. 1996) which can be regarded as 'one of the most complex and uniquely human of cognitive activities' (Van Den Broek et al. 1996), and thus, understanding the nature of comprehension is far from trivial.

One area that offers a valid means of investigating comprehension is that of Discourse Process. Bamberg and Moissinac (2003) define discourse as 'broadly taken to mean the use of language beyond that of a single sentence'. Discourse Process analyses the way in which sequences of sentences combine to produce coherent sections of language, and thus extends the traditional linguistic study of the construction of individual sentences (Crystal 1997; Graesser et al. 1997). Requirements techniques aim to communicate sets of concepts and meanings which have been constructed, often within sizeable documents, to various stakeholders in a system. Moreover, they typically capture the interaction between an actor and the system in order to accomplish the actor's goal across a number of statements (Cockburn 2001; Kulak and Guiney 2000). Therefore, discourse process is particularly relevant for the understanding, and improvement, of requirements.

The aim of this paper is to investigate the lessons that can be learned from the field of Discourse Process, particularly about the nature of comprehension, for improving the writing of requirements and specifications. Further work, outside the scope of this paper, examines the implication of these ideas for existing requirements authoring guidelines. However, our intention here is, by thorough scrutiny of discourse process, to highlight the issues and provide lessons learned. The following Section 2 introduces the principles of discourse process, Section 3 describes the application of discourse process theory, Sections 4 and 5 discuss the roles of background knowledge and of the reader, and Section 6 describes limitations of the approach. Section 7 gives an overview of current use case writing guidelines and some examples, before Section 8 makes proposals for their refinement in the light of discourse process theory and, finally, Section 9 offers some conclusions.

2 General principles of discourse process

As people read and comprehend, they build a mental representation or model of the text and of the situations portrayed within the text. This mental representation consists of multiple levels of code in memory. Of the various levels of representation that have been proposed, most discourse psychologists accept three: surface code, textbase and situation model. The surface code consists of the exact wording and grammar of clauses, i.e. it preserves the actual text. The textbase preserves the meaning of the explicitly stated information in the text but not its exact wording and grammar. It is usually viewed as consisting of a structured set of text propositions. Each proposition refers to a state, event or action and may possess a truth value. It contains a predicate (e.g. a main verb, adverb, adjective or connective) and one or more arguments (e.g. nouns or other propositions embedded in it). The predicate denotes a relation between arguments. Each argument plays a functional role, such as agent (the entity that causes the action of the verb), patient (the entity that undergoes the action of the verb), instrument or location.

The propositions form a network, being linked to each other through shared arguments and other connectives in the text. They typically form hierarchical structures. As more text is read, new propositions are added to the network, forming a coherent knowledge structure. The textbase represents a superficial understanding and enables such tasks as recall of the text or the answering of questions which only require information from one sentence.

The situation model is constructed from inferences (additional propositions, Foltz 2003) made by the reader as they extract the meaning from the text and activate relevant background world knowledge from their memory stores. Almost every aspect of comprehension depends, at least partly, on these inferences constructed by the reader. These are necessary for filling in gaps and thus building the situation model and for increasing coherence. There are two types, bridging inferences and elaborative inferences. Bridging inferences link the current clause to the preceding text, thus maintaining local text coherence. Elaborative inferences are ideas that are strongly implied by the context of a discourse; they supply additional propositions based on world knowledge and elaborate the text which is being read, but do not affect coherence (Zwaan and Singer 2003). The situation model, therefore, represents deeper understanding and enables the connection of two or more sentences of the text, the linking of information from the text with background knowledge and problem-solving. All of these levels contribute to the meaning representations that readers mentally build as they comprehend.

2.1 Coherence

If a text is to be comprehensible, it must be coherent, i.e. the reader must be able to identify relations among the ideas in the text (Zwaan and Singer 2003). For example, many researchers have argued that the presence of expressions in a text referring to the same entities (coreference) promotes coherence (Gordon et al. 1993). Two main types of coherence exist: local and global. Local coherence occurs when each sentence is related to the next, e.g. through argument overlap (i.e. the incoming statement of a text contains a noun or pronoun which refers to the same entity as a noun or pronoun in the preceding sentence). Global coherence requires that larger sections of text are clearly related to each other and to the overall topic. For Graesser et al. (1997), global coherence is achieved if the reader can connect the incoming statement with the overall structure of the text or with information from the text which is no longer held in working memory. Gibbs (1996), however, states that cognitive psychologists have argued that the comprehension of text is driven by the reader's search for cause and effect and that these causal relations provide much of the global coherence of memory. Graesser et al. (1997) similarly take the view that readers actively seek causal explanations of why events in the situation model occur and why the writer conveys information as they are reading. McNamara et al. (1996) also refer to explanatory coherence which is achieved when the content of the text supplies background knowledge which the reader may lack but which is needed to understand the text. Studies have demonstrated that increases in the local, global and explanatory coherence of texts have resulted in substantial increases in text recall by readers (Britton and Gulgoz 1991).

2.2 Advantage of first mention

The Advantage of First Mention proposes that the first entity mentioned in a clause, regardless of grammatical position or semantic role, enjoys a higher level of activation and is accessed more quickly from memory than the other entities. This is explained by the Structure Building Framework (Gernsbacher 1996) which is described below.

2.3 Structure building framework

According to the Structure Building Framework, the aim of comprehension is to create cohesive mental representations, or structures. When readers build a mental representation of the text, they perform three processes. First, they lay a foundation when a new topic is introduced by setting up the mental structures needed to establish coherence. Aaronson and Ferres (cited by Foltz 2003; Gernsbacher 1996) argue that readers spend more time processing the first word of a clause or sentence. This is attributed to the time taken to generate an initial framework for processing the new discourse. They then build upon this foundation, if possible, by mapping any information from the subsequent text which is relevant to the same topic onto the structures that have been developed. If the subsequent text is not relevant, however, the reader shifts attention and builds a new substructure. These processes produce a branching structure which is an overall cohesive mental representation of the text.

2.4 Event-Indexing Model

In Zwaan and Singer's Event-Indexing Model (2003), the reader of simple stories, or literary short stories, monitors five conceptual dimensions in order to construct a multi-threaded situation model. The five dimensions are protagonists, causality, protagonists' goals, geographic space and time. When the reader processes a new statement, a break in continuity may occur on any of these dimensions. Zwaan and Singer (2003) found that the time taken to read a particular event in a story increased as a function of the number of these dimensions which contained a discontinuity (i.e. lacked coherence). Each dimension had a unique impact on reading time. Therefore, the Event-Indexing Model recommends that the writer should strive, wherever possible, to avoid a break in continuity on any of these dimensions.

2.5 NVN strategy

The NVN (Noun Verb Noun) strategy is one of the most important heuristics in comprehension (Ferreira 2003). The reader is inclined to assume that the subject of a sentence is also the agent of some action, and this can be seen in much guidance (such as for use case descriptions) which suggests adopting, subject, verb and object structures.

3 Application of discourse process theories

3.1 Pronouns

There are various discourse process theories regarding pronouns. These theories include: Gernsbacher's Advantage of First Mention (1996), "centering theory" (discussed below) and the subject assignment strategy, which states that if a pronoun is ambiguous, the reader is likely to interpret it as referring to the subject of the preceding sentence (Gordon and Scearce 1995). In addition, the theories of (a) Matthews and Chodorow and (b) Sidner (both cited by Greene et al. 1992) argue that an entity, which is the grammatical subject of a sentence, is a strong candidate for identification with a subsequent pronoun which is also in the subject position. If these theories are accepted, then the success of a pronoun depends not just upon the pronoun itself but also on the context of its referent. Therefore, the writer who wishes to reduce the risk of ambiguity associated with a pronoun must take care to ensure that the referent of the pronoun is introduced in the subject position.

Certain discourse process theories support the use of pronouns, however. These include the theory that the processes by which the referent of a pronoun is identified actually increase the accessibility in memory of the intended referent (Greene et al. 1992). Once again, opposing theories exist; (Gernsbacher 1996), for example, argues that all anaphoric references increase accessibility. McKoon et al. (1996) also reject the theory that pronouns increase accessibility, proposing instead the theory of memory-based text processing, which is based on the principle that general memory processes make the intended referents available to the reader even before the pronouns themselves are used.

The use of pronouns can sometimes avoid awkwardness, but this is a factor which is largely based on intuition rather than empirical evidence. Tanenhaus et al. and Tanenhaus and Carlson (both cited in Gordon and Chan 1995) demonstrate that certain kinds of references to entities seem to be understood as a result of the entities and references sharing the same thematic roles across sentences. In such circumstances, an explicit reference to the entity, e.g. repeating the noun rather than using a pronoun, results in a text which is awkward to read. Thus, there is a trade-off between avoiding awkwardness and its accompanying increase in reading time on the one hand and avoiding ambiguity on the other. However, increases in reading time are of minor importance, often being measured in milliseconds, while the importance of avoiding the risk of ambiguity is paramount, and particularly important within our context of requirements and specification.

In summary, the writer should consider carefully where to use pronouns. There is a trade-off between the benefits of increased coherence on the one hand and the increased risk of ambiguity coupled with increased time and effort on the other. Furthermore, although Gordon et al. (1993) demonstrate that pronouns can affect reading time, they also indicate that the use of pronouns has no significant effects on readers' accuracy when answering comprehension questions. This has clear implications for requirements engineering, where accuracy of interpretation far outweighs minor differences in the speed of comprehension. Pragmatically, it would therefore be wiser for the writer to avoid using them, given the ambiguity that they can create, with little apparent gain for the reader.

3.2 Use of the passive voice

The disadvantages of using the passive voice are demonstrated by many researchers in the fields of linguistics and discourse process. A number of researchers demonstrate (cited by Gordon and Chan 1995) that when sentences are presented in isolation, passive versions that can be changed into active voice without losing their meaning, take longer to understand than their active form. Similarly, the results of an experiment performed by Gordon and Chan (1995) showed that passive sentences were read more slowly than active sentences which had a similar meaning. Indeed, the increase in time taken is not just limited to reading and understanding sentences; experiments performed by Ferreira (2003) demonstrate that passive sentences also take longer to formulate than active ones. Hence, there is a cost for the author as well as the reader where the passive is concerned.

Of greater importance than the speed of writing and reading is the accuracy of understanding. Ferreira (2003) states that evidence suggests that syntactic complexity increases the chances of misinterpreting text. Her own experiments indicate that comprehenders performing the task of identifying agents and patients in active and passive versions of the same sentences not only take longer but are less accurate when answering questions about the passive forms. Indeed, Ferreira's findings consistently demonstrate that the passive form is more difficult to understand than its active counterpart. The results of her experiments indicate that even non-reversible passive sentences may not be particularly easy to understand, although previous researchers have claimed that this is the case (Harley 2001). (A non-reversible sentence is one in which the subject and object cannot be transposed without the sentence becoming implausible.)

The major benefit of the passive is that it allows an entity which is not the thematic agent to be the grammatical subject of a sentence. The importance of the grammatical subject and of the first mentioned entity in a sentence is stressed by various theories and studies in discourse process and linguistics. Clark (cited by Johnson-Laird 1968) claims, for example, that 'people put what they want to talk about…in the beginning of the sentence' and Ferreira (1994) notes that there was a trend for participants in her tests to be more accurate with the first role that they encountered in a sentence.

In summary, for the context of requirements documents, it appears that there is an increased risk of misinterpretation when using the passive voice, which could lead to problems with understanding of specification and lead to design errors; therefore, avoidance of the passive appears to be the safest option.

3.3 Simplicity

Where the aim is to increase the ease of comprehension, it seems sensible to avoid the use of complex structures, such as the passive voice, the negative form, and adjectives and adverbs. Moreover, this is supported by discourse process theories and studies which indicate that complexity increases reading time (e.g. Kintsch and Van Dijk 1978; Ferreira 2003) and that challenging structures are sometimes misinterpreted. However, even the concept of simplicity is complicated; for example, Ferreira (1994) noted that studies exist which demonstrate that comprehenders are generally willing to accept wildly implausible meanings of simple sentences. Moreover, good use of discourse cues, e.g. words or phrases, such as "because", "first", "although" and "also", which signal grammatical and semantic relationships between entities in a discourse, improves comprehension and recall. Although, as a caveat, indiscriminate use of semantic cues may have damaging effects on the reader's ability to recall the text (Zwaan and Singer 2003). Evidence suggests that how beneficial discourse cues are to readers may depend upon their reading skills or level of prior domain knowledge (Moore and Weimer-Hastings 2003). The situation is further complicated by the problem that whether a concept is new, or not, is not just dependent on the preceding text; it is also dependent on the reader's knowledge. Thus, what is new to one reader may not be new to another.

Therefore, there is a trade-off between simple sentences which can result in reduced reading time and decreased risk of misinterpretation and more complicated structures which can also lead to reduced comprehension time and decreased risk of misunderstanding. Simple structures may not be as beneficial to some readers, i.e. those with poorer background knowledge, as they are to others. Again, this appears to have significant implications for requirements guidelines, where, for example, the majority of use case description guidelines advocate extremely simple structures at all times. The problem being that this may not always be beneficial to comprehension.

4 The role of background knowledge

References to the importance of the reader's background knowledge occur throughout the discourse process literature (Mannes and St. George 1996). Gordon and Scearce (1995) argue that knowledge-based processes play a very early role in the processing of language, while Kintsch and van Dijk (1978) note that 'comprehension always involves knowledge use and inference processes' and that the reader's knowledge largely determines the meaning derived from a text.

Graesser et al. (2003) describe the knowledge-based inferences which readers generate during narrative comprehension as 'critical building blocks' and suggest that the higher the number of levels of representation constructed and the higher the number of inferences generated by the reader at each level, the greater the improvement in comprehension. However, the generation of novel knowledge-based inferences is believed to place more demands on working memory. The greater the demands that a potential inference places on working memory, the lower the likelihood that it will be generated during comprehension. Thus, the more familiar the relevant background knowledge is to the reader, the more likely he or she is to generate inferences from it. Again, this would seem to suggest that background knowledge of the subject area is extremely important to the software engineer; in understanding the item under scrutiny, a view that is borne out by much software engineering literature (e.g. Curtis et al. 1988, 1989).

4.1 Coherence and background knowledge

The interaction between the background knowledge of the reader and the coherence of the text is investigated by McNamara et al. (1996). Their findings show that where recall of the text is concerned, i.e. just shallow understanding, coherence benefits readers' understanding, at least on a superficial level, but that domain knowledge can compensate for lack of coherence. They attribute this contradiction, between increased coherence both facilitating comprehension and inhibiting understanding and learning, to the different levels which exist in the mental representation of the text generated by the reader. Readers are able to form a good textbase if the text is fully coherent. However, if the text lacks coherence at both local and global levels, readers are forced to fill the gaps themselves with a bridging inference. If they are unable to do so, they may be forced to build a situation model. This prevents them from basing their understanding purely on the superficial textbase. As a result, it enables them to be particularly successful in answering questions which require bridging inferences or problem-solving and in a sorting task. Coherence and domain knowledge can, therefore, have different effects on understanding, depending on the different level of mental representation being constructed and on the different nature of the tasks being performed by the reader.

Although increasing the coherence of a text will benefit readers with poor background knowledge, it may actually disadvantage readers with good background knowledge since it may reduce the amount of active processing that they perform during comprehension. Studies have shown that, generally, people are more likely to remember information which they have actively generated themselves than information which is simply presented to them without effort on their part. They are also more able to use information which they have generated themselves in new situations. Mannes and Kintsch (1987) argue that the active generation of inferences during reading can be beneficial, if the inferences are successful, since it results in the creation of more links between the new information and the information which already resides in the reader's own knowledge base. They associate more active processing with the construction of a more detailed situation model and with superior performance on a problem-solving task. Therefore, McNamara et al. (1996) believe that it is sometimes better to allow readers to construct their own coherence rather than let the writer construct it for them in the text, although they add that it may be necessary to supply exactly the right amount of difficulty in a text to enable greater understanding by readers with high knowledge, making the text neither too easy nor too difficult. The degree of domain knowledge possessed by the reader is important. That is, the same text cannot be equally suitable for every reader.

5 The reader in comprehension

The lessons which may be learned from discourse process theories are also affected by the sheer complexity of the process, or rather processes, of comprehension and by the differences between individual readers (Goldman et al. 1996). For example, research suggests that the selection of propositions which remain in short-term memory (which approximately holds the most recent clause being comprehended) at the end of a sentence is influenced by the reader's goals (Fletcher et al. 1996). Kintsch and van Dijk (1978) also comment on different readers setting up different goals and proposing that the special purpose of the reader overrides the text structure. According to Van den Broek et al. (1996), many researchers share the view that readers set for themselves a goal or standard for coherence. They then generate the amount of inferences necessary to attain this standard.

Thus, the discourse literature strongly indicates that the reader, as well as the text, is an important factor in understanding. The success of comprehension depends upon the extent to which there is harmony among the author's intended meaning of the text, the explicit text, and the reader's constructed meaning of the text (Graesser et al. 1994).

6 Limitations of discourse process

Discourse process is a comparatively young field (Graesser et al. 2003), and its researchers have proposed various theories which are often in opposition to one another (Gordon et al. 1993). Moreover, when theories and models are tested, there is a strong tendency for the results to 'suggest' or 'indicate' findings rather than unequivocally validate them. Graesser et al. (2003), for example, discuss the need or the inevitability of integrating neuroscience into discourse process research and note that a few studies have already made use of functional brain imaging techniques to examine the activity of the brain during discourse comprehension. Furthermore, the traditional assumption that underlies many discourse process theories—that the reader constructs networks of propositions during comprehension—is now challenged: Stanfield and Zwaan (2001) and Zwaan et al.(2002), among others, propose instead that during language comprehension, people activate perceptual symbols of referents. In addition, the theories indicate that there is often a trade-off between the potential benefits and disadvantages of certain linguistic forms, such as complex structures which can result in both increased reading time and risk of misunderstanding and decreased reading time and risk of misunderstanding, depending on the circumstances.

Nevertheless, there are still significant lessons to be learned from discourse process which have important implications for the writing of requirements. These include the importance of the reader's background knowledge, the existence of different levels of understanding, the need to make the reader do active work, the avoidance of pronouns and passive voice, and so on, all of which appear to have direct implications for the documents produced within the requirements phase.

7 A selection of use case writing guidelines

Having examined discourse process and its implications for comprehension, it is appropriate to consider the extent to which current guidance, in particular for the production of use case descriptions (which may be viewed as the vital component of the use case specification), is supported by the findings of discourse process research. A comprehensive review of a number of approaches was undertaken (Adlem 2007) including CP rules (Cox et al. 2001), CREWS authoring guidelines (Ben Achour et al. 1999), various template approaches, Cockburn's rules (Cockburn 2001) and other advice within software engineering, each rule for each set of guidelines was analysed and discussed, and the empirical evidence to support such rules also evaluated (Phalp et al. 2010).

A flavour of this work can be given, in that we show application of the approach to a selection of rules (including both CREWS and CP rules), which serves to bring a software engineering (use case) context to the work. The main findings of this work were then used to inspire a further set of proposals for templates, rules and guidelines. The complete description of the justification for each rule is large and extensive; hence, in order to present the work with sufficient brevity, we have chosen to tabulate the rules, their justification and derivation or antecedence, and this paper gives a snapshot from the work (Phalp et al. 2010).

7.1 The CP use case writing rules

The CP Use Case Writing Rules (Cox et al. 2001; Phalp and Cox 2002; Phalp et al. 2007a) consist of a set of eight Style Rules and a set of three structure rules. As we shall note later, the distinction is somewhat artificial, but broadly structure rules guide the author overall as to the grammatical structures to be adopted across all of the descriptions, while Style Rules tend to offer guidance, typically at the level of the sentence, which may be applied case by case, dependent on the content of the particular clause or construction.

A few examples are outlined here, both to illustrate the relation to the underlying discourse process theories described in the previous section and to illustrate the general approach.

CP Style Rule 2 *Avoid pronouns if there is more than one actor.*

Examples:

- *Patient stands next to the doctor.*
- *He puts the prescription in his pocket.*
- Who is "he"? Whose pocket is "his"? Write proper nouns/names instead:

- *Doctor puts the prescription in the patient's pocket.*
- *GP puts the prescription in the customer's pocket.*
- This sentence is also at fault because it uses synonyms (GP for doctor and customer for patient). Only use the agreed language of the domain since a synonym does not convey the same meaning.

The major argument against the use of pronouns in use cases is the risk of ambiguity which they may introduce. More than one entity in a sentence could share the same gender and number and could reasonably perform the action of the verb in the clause containing the pronoun. Accordingly, each entity could be a potential candidate for identification with the pronoun. An unheralded pronoun in a use case description is even more likely to be ambiguous than a pronoun whose intended referent has been mentioned in the text immediately preceding it. Associating the wrong entity with a pronoun in a use case description or other requirements document could prove to be a very time-consuming and costly mistake.

Other arguments against the use of pronouns in use cases include the fact that a pronoun in the singular, e.g. "he" or "she", attributes gender to an actor, detail which is too specific and may be inappropriate. The pronoun "it" does not suffer from this problem but instead it could lead to confusion between the system under discussion and another system which is an actor in the use case.

As discussed earlier, many discourse process theories exist which involve pronouns. Various theories suggest that the use of pronouns is beneficial to comprehension and thus oppose CP Style Rule 2. These include the theory that the processes by which a pronoun is resolved actually leave the intended referent of the pronoun in a more accessible state than other potential referents (Greene et al. 1992), thus making it easier for the reader to retrieve it from memory.

Discourse process offers no conclusive arguments either for or against the use of pronouns in use cases and the situations in which pronouns may offer benefits often impose constraints and extra work on the part of the use case writer. Therefore, given the ambiguity and possible errors that pronouns can create, it is right to recommend that pronouns should be avoided in use cases. However, CP Style Rule 2 needs to be more stringent and to advocate that pronouns should be avoided even if there is only one actor. Even in use cases where there is only one actor, the use of pronouns is likely to raise the problem of gender (if the actor is human) and the problem of ambiguity (if the actor is another system since the actor may be confused with the system under discussion).

CP Style Rule 3 *No adverbs or adjectives.*

Avoid using adverbs and adjectives, these add unnecessary clutter to the description and give values that are difficult to quantify. Only use negatives in alternative and exceptional flows of events. Avoid using pronouns (style 2) (e.g. he, she, it, we, their etc.) and synonyms.

Examples:

- *Doctor writes the prescription slowly.*
  - *slowly* is an adverb—we do not need to know how the doctor writes the prescription, just that *the doctor writes the prescription.*
- *Patient swallows the big pill*
  - *big* is an adjective and is unnecessary; you should write *the patient swallows the pill.*

As is the case with CP Style Rule 2 regarding pronouns, discourse process offers conflicting theories where the use of adverbs and adjectives are concerned. Sentences containing adverbs and adjectives are complex structures. Complex structures increase reading time (Kintsch and Van Dijk 1978; Zwaan and Singer 2003) and are sometimes misinterpreted (Ferreira 2003). However, Pragmatic Principle number 1: ‘Something new should be signalled syntactically and embellished, e.g. with adjectives’ (Graesser et al. 1997) suggests that there are situations in which adjectives and adverbs can aid the comprehension of use cases. Thus, there is a trade-off between simple structures avoiding increased reading time and increased risk of misinterpretation, and complex structures containing adjectives or adverbs achieving the same effect. Given this trade-off, it appears to be wiser for the use case writer to avoid complex structures unless there is a specific benefit to be gained from them. Once again, context is important and where adjectives and adverbs should be used is constrained: they are to be avoided unless they are being used in connection with new information. This raises the difficult question of the use case reader’s background knowledge—what is new to one reader, or type of reader, may already be known to another. Moreover, it places the onus of monitoring or being aware of the reader’s knowledge on the use case writer.

It, therefore, appears that CP Style Rule 3 is correct in certain circumstances but not in others, depending on the reader’s background knowledge. It may be suitable for readers who possess high background knowledge, such as domain experts and end-users, but less appropriate for those with lower domain knowledge, such as designers. This is supported by Phalp et al. (2007a) who argue that although adjectives and adverbs may introduce subjectivity in a use case which can encourage misunderstanding, adjectives can aid designers since they may be identified as services within objects. Thus, CP Style Rule 3 may be suitable for requirements but not for specification and, in particular, design. However, the need for accuracy in specification and design may outweigh the benefits of embellishing text for readers with poorer domain knowledge. For example, the use of an adverb such as “quickly” as opposed to the phrase “after one minute” greatly reduces accuracy. It therefore seems sensible to apply CP Style Rule 3 to specification as well as requirements but to modify it for design so that adverbs are avoided but adjectives can be included in use cases.

This suggests that it is beneficial to write different use cases for different types of readers, such as end-users and designers. In which case, different authoring guidelines would also be required for the different types of use cases. This contrasts with Phalp et al. (2007b) who recommend the pragmatic approach of retaining natural language but providing guidance to use case writers so that use case descriptions are both comprehensible to a non-technical audience and sufficiently rigorous for specification.

CP Style Rule 4 *Avoid negatives.*

Negatives are another example of complex structures and as such can lead to increased reading time and misinterpretation. Thus, CP Style Rule 4 is supported by discourse process. However, Phalp et al. (2007a) state that negatives should only be used in use cases in alternative and exceptional flows. This seems logical since the main flow should describe the shortest successful path through the use case and therefore should not require negatives. In contrast, the alternative and exceptional flows deal with potentially negative conditions, namely those situations where the main flow of events is not appropriate for a particular set of circumstances or it fails. This is an example of a rule which applies to some sections of a use case but not to others. It should therefore be amended to ‘Avoid negatives except in alternative and exceptional flows’.

7.2 CP and CREWS structure rules

Structure Rule 1 *Subject verb object. ('Verb' refers to present simple tense).*

Example: Operator presses the button.
CP Structure Rule 1: Subject verb object. ('Verb' refers to present simple tense and assumes active voice).
Example: Operator presses the button
CG1: <agent> <'move' action> <object> from <source> to <destination>;
Example: Clerk sends the report from the store to the office.
CG2: <source agent> <'put' action> <object> to <destination agent>;
Example: Clerk gives the report to the manager.
CG3: <destination agent> <'takes' action> <object> from <source agent>;
Example: Manager gets the report from the clerk.

Structure Rule 1 advocates a very simple grammatical structure. This is supported by the discourse process theories which hold that complex structures increase reading time. Kintsch and Van Dijk (1978) predict that increasing the complexity of the surface code (the actual words of the text) without changing the underlying meaning is expected to decrease the amount of text a reader can process at a time. However, Zwaan and Singer (2003) argue that text reading time increases systematically with the number of propositions that it contains (the number of words being held constant) and with the number of different arguments in the text (the number of propositions being held constant). In addition, complex structures are also sometimes misinterpreted (Ferreira 2003).

It is assumed that Structure Rule 1 recommends that the verb is in the active voice; this is implied since it does not mention a preposition and the preposition "by" would be required before the agent in a passive sentence. In which case, it is also supported by the NVN (Noun Verb Noun) strategy which is a very simple but extremely important heuristic in comprehension. Owing to the vast number of sentences in the English language which conform to the pattern of agent before patient, the reader is strongly inclined to assume that the subject of a sentence is also the agent of an action and that the object is the patient. Ferreira's (2003) findings suggest that sentences which violate the NVN strategy are more difficult to understand than those which comply with it. Therefore, Structure Rule 1 aids comprehension by satisfying the reader's expectations regarding word order.

8 Proposals for rules and guidelines

Having investigated those discourse process theories which have implications for requirements and their agreement with guidelines for use case authoring, we find that different use case structures are required for different phases of the software engineering process. Therefore, various qualities necessary for use cases intended for requirements, specification and design have been identified from the discourse process theories and organised according to phase (summarised in Table 1).

Having understood which qualities are required and for which phase, we then consider how guidelines can best produce, in the use case description, those qualities which are desirable for comprehension. This ultimately leads to the production of our own set of use case rules, which again draw upon both discourse process and existing software engineering, and particularly use case, research, but which for brevity are presented in summary form within Table 2 (which owing to its size is given within an appendix).

Table 1 Desirable qualities of use cases intended for requirements, specification and design

Phase/desirable qualities of use cases	Req's	Spec	Des'n	Sources
Standard format	✔	✔	✔	Anda et al.'s (2001) template guidelines; Kulak and Guiney (2000), Cockburn (2001), Schneider and Winters (2001)
Completeness	✔	✔	✔	7 C's Facet 1.1; 7 C's Facet 2.2; Phalp et al. (2007b), Cockburn (1995), Ben Achour et al. (1999), Kulak and Guiney (2000), Schneider and Winters (2001)
Conciseness	✔	✔	✔	Pragmatic principle number 5; Event-indexing model; 7 C's Facet 1.2; Phalp et al. 2007a, Cockburn (1995), Kulak and Guiney (2000)
Accuracy	✔	✔	✔	Kulak and Guiney (2000), Schneider and Winters (2001), Phalp et al. (2007b)
Logic	✔	✔	✔	CP Style Rule 7; CP Style Rule 8; 7 C's Facet 2.1; 7 C's Facet 2.3; 7 C's Facet 3: Phalp et al. (2007a)
Coherence	x	✔	✔	Various discourse process theories, including event-indexing model; McNamara et al. (1996) CP Style Rule 7;7 C's Facet 3
Appropriate level of detail	✔	✔	✔	Anda et al. (2001), Cockburn (2001)
Consistent level of abstraction	✔	✔	x	7 C's Facet 4 (2007b); Cox et al. (2004)
Readability	✔	✔	✔	7 C's Facet 5.1 (2007b); Cockburn (2001), Schneider and Winters (2001)
Natural language (as opposed to program-like)	✔	x	x	Cockburn (2001), Kulak and Guiney (2000)
Embellishment (e.g. with adverbs, adjectives)	x	x	✔ (adj OK)	Pragmatic Principle number 1; Phalp et al. (2007a)

However, we first consider the nature of guidance, in terms of which aspects should be considered genuinely as rules and which should be subject to rather looser 'guidance' (terms which are often used interchangeably).

8.1 Rules and guidelines

The distinction between Style and Structure or Content Rules, as found in the CP Rules and CREWS Guidelines, appears to be unnecessary. Both types of guidance are concerned with *how* the information is presented, not *what* information is included in a use case. The term "Content", as used in the CREWS Guidelines, is misleading since it implies what is included in the use case. The term "Structure", which is used instead in the CP Rules, is preferable but is still not ideal. It refers more to sentence structure or composition rather than the structure of the use case. Moreover, it is sometimes difficult to decide whether a rule is a Style or a Structure/Content Rule. For example, some of the CREWS Content Guidelines (Ben Achour et al. 1999) previously formed part of the Style Guidelines (Rolland and Ben Achour 1998).

Nevertheless, two levels of guidance may be desirable: rules, which are enforceable and must be obeyed and guidelines, which indicate an ideal that cannot always be followed. This distinction is made in order that the use case author does not omit any rules which can and must be applied but, conversely, does not spend too much time trying to apply a guideline which is not always practicable. However, a possible danger of this approach is that guidance which is particularly important but is not always enforceable will appear as a guideline rather than a rule and may therefore be perceived as less important.

8.2 Rules for use case descriptions

As noted above, Table 2 (given as an "Appendix") summarises the proposed rules, the theories and arguments which support them and the readership for whom they are applicable. Clearly, while Table 2 represents a summation of the antecedence and justification of each rule, much of this substantial theoretical underpinning may be of little interest to the practitioner. Hence, in keeping with the desire to consider a variety of audiences Table 3 (included below) gives a brief 'hopefully readable' description of each rule and also suggests where (that is at what phase) the rule is most appropriate.).

The differences which have been identified between rules for use cases intended for different tasks are few in number but are important. The proposed Rules have all been considered applicable to use cases intended for requirements. However, one of them, R13: 'Avoid the use of structures which resemble programming language, e.g. "if-then-else" statements and "end loop"', is not necessary for use cases intended for specification.

Table 3 Rules which apply to requirements, specification and design

	Rule	Req	Spec	Design
R1	Each use case should conform to a standard template appropriate to the organisation and project	✓	✓	✓
R2	The use case should contain all the information that is required to answer the problem but no information that is irrelevant to the problem statement	✓	✓	✓
R3	The use case should follow a logical path with events in the description in the correct order	✓	✓	✓
R4	The use case should complete as an end-to-end transaction (which can include alternative/exceptional flows) with appropriate post-conditions satisfied	✓	✓	✓
R5	The logic of the use case description should provide a rational answer to the problem	✓	✓	✓
R6	When an action occurs, there should be a meaningful response to that action	✓	✓	✓
R7	Show the actor's intent, not the actor's movements in operating the system's user interface	✓	✓	✓
R8	The use case should be at a consistent level of abstraction throughout	✓	✓	x
R9	Each sentence in the description should be on a new, numbered line	✓	✓	✓
R10	Alternatives and exceptions should be described in a section below the main description and the sentence numbers should agree	✓	✓	✓
R11	Use simple present tense throughout	✓	✓	✓
R12	Use the active not the passive voice	✓	✓	✓
R13	Avoid the use of structures which resemble programming language, e.g. "if-then-else" statements and "end loop"	✓	x	x
R14	Each event should be written in the form "Subject… verb… object… (optional:) prepositional phrase"	✓	✓	✓
R15	Avoid pronouns, synonyms and homonyms	✓	✓	✓
R16	Avoid adverbs and adjectives (for design avoid adverbs only)	✓	✓	x
R17	Avoid negatives except in alternative and exceptional flows	✓	✓	✓
R18	Underline other use case names referenced in a use case	✓	✓	✓

Table 4 Guidelines which apply to requirements, specification and design

	Guidelines	Req	Spec	Design
G1	There should be logical coherence throughout the description. The sentence being written should repeat a noun in the last sentence or a previous sentence, if possible	x	✔	✔
G2	Where possible ensure that the most important entities are placed at the beginning of the use case or as the first mentioned entity and/or the subject of an event	✔	✔	✔
G3	Do not mention an actor in an event who is different from the actor(s) in the previous event, if possible	x	✔	✔
G4	If the new event occurs much later in time, consider whether it should be a separate use case	x	✔	✔
G5	If the new event occurs in a different spatial setting, consider whether it should be a separate use case	x	✔	✔
G6	Give explanations if necessary but place them in a footnote	✔	✔	✔

Although it could be argued that R13 is useful for all three phases since it is desirable to avoid unduly influencing the later phases of the process, the main intention behind the rule is to prevent structures which resemble code from being confusing and off-putting to non-technical readers. The difference between the rules for requirements and design, however, is more marked. Not only is R13 considered not necessary for use cases for design, but R8: 'The use case should be at a consistent level of abstraction throughout' has also been omitted while R16: 'Avoid adverbs and adjectives' has been modified to 'Avoid adverbs' on the grounds that mixing abstraction levels and the presence of adjectives may benefit the design process.

8.3 Guidelines for use case descriptions

In addition to the rules described above, a number of guidelines are also given. For a full summary of the proposed guidelines, the theories and arguments which support them, the guidelines and guidance which are in agreement and the readership for whom they are applicable the reader is referred to Phalp et.al. (2010). However, Table 4 gives a brief description of each guideline and also suggests where (that is, at what phase) the guideline is most appropriate.

9 Conclusions

The field of discourse process is often used as a theoretical underpinning for assertions about the nature of comprehension and its implications for software engineering. However, few authors have attempted to unravel the complexities of comprehension before moving to its implications for software quality. In contrast, this paper sought to examine carefully the literature and lessons learned from discourse process in order to understand more fully the implications, particularly for the requirements phase.

The authors have found that there are many implications for the production of requirements, or indeed, any software engineering documentation. For example, perhaps the most intriguing finding is that despite the drive for ease of comprehension, it appears that it is not always beneficial for the reader if the author makes the text as explicit and coherent as possible. That is, a deeper understanding can be engendered where the reader needs to carry out work themselves and construct their own explanations rather than for the text to present them with everything too easily. This is particularly so where the reader has a high level of background knowledge and is thus able to provide suitable explanations. This implies that a text should be challenging enough to encourage active processing by the reader but not so difficult as to become incomprehensible. Similar findings relate to the simplicity of text, the use of pronouns, and the use of the passive voice. For example, while there may be some gains in the speed in which very simple text, or text with pronouns, can be parsed, such advantages are outweighed by the increased understanding, or lack of ambiguity gained by not using pronouns, or by having text that is neither too simple nor overly complex.

Of particular importance for our work is the implication that different audiences react in different ways to the text that they are given, depending on a number of factors, such as their experience, familiarity with the subject matter and so on. Hence, for software engineering that different structures, and the guidance used to produce them, might be required for different audiences (end-users, clients, analysts, designers) and different tasks (elicitation, requirements validation, specification, design). Therefore, this paper proposes different sets of rules and guidelines according to the level of domain knowledge of the use case readers, which is expected to be high for domain experts and end-users and lower for developers engaged in specification and design. The proposed rules and guidelines also take into account the different nature of the tasks involved in the different phases of the software engineering process. Thus, requirements analysis involves developing a richer understanding of the problem domain, while accuracy of information is particularly important for specification. Moreover, consistency of abstraction is viewed as a desirable quality in use cases intended to be used for requirements and specification but not in use cases intended for design since it has been shown that designers think and move through different levels of abstraction when they consider a design problem.

It was considered sensible to divide the proposed guidelines into rules, which can always be applied to use cases and are therefore viewed as compulsory, and Guidelines which offer guidance which is desirable but not always feasible and are therefore to be followed whenever practicable. Eighteen basic rules were identified but only six basic guidelines. It was found that separate sets of the proposed Rules were necessary for use cases intended for each of the three phases, requirements, specification and design. In contrast, separate sets of the proposed Guidelines were necessary for requirements on the one hand and for specification and design on the other.

Although within this paper our analysis has focussed on use cases, the lessons about comprehension also apply to requirements documentation in general since requirements documents are also a text form. Indeed, they could be applied further still, e.g. to the field of teaching software engineering. Where those being taught possess a high level of background knowledge, ensuring that any textual material is not too coherent (by such means as removing connective words which indicate the relation between sentences or ideas and replacing words to reduce argument overlap) will encourage more active processing and a deeper understanding. However, this is not easily achieved. McNamara et al. (1996) believe that it may be necessary to provide exactly the right amount of difficulty in a text in order to achieve greater understanding by readers with high knowledge. If the text is too difficult, the reader may not be able to construct coherence. If on the other hand, it is too easy, the benefits gained by more active processing may be too small. A text should be challenging enough to encourage active processing by the reader but not so difficult as to become incomprehensible. While difficulties should be introduced into texts for readers with high knowledge, they should be introduced carefully. The same text cannot be equally suitable for every reader. As Kintsch and van Dijk (1978, p. 372) state, readability "cannot be considered a property of texts alone, but one of the text–reader interaction".

Appendix

See Table 2.

Table 2 Proposed rules and the theories which support them

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R1	Each use case should conform to a standard template appropriate to the organisation and project		Ensures standardization; ensures important elements included; easy to understand; can be created to fit organisation/project	Anda et al.'s template guidelines (2001), Cockburn (2001), Kulak and Guiney (2000), Schneider and Winters (2001)	All types of reader
R2	The use case should contain all the information that is required to answer the problem but no information that is irrelevant to the problem statement	Pragmatic principle number 5: incoming sentence should be relevant; Event-Indexing Model: important to avoid intentional discontinuity		7 C's Facet 1 (Phalp et al. 2007b)	All types of reader
R3	The use case should follow a logical path with events in the description in the correct order	Structure building framework; pragmatic principle number 6: order of mentioning events should correspond to chronological order of events in the situation model; Event-Indexing Model: important to avoid temporal discontinuity		7 C's Facet 2 (Phalp et al. 2007b)	All types of reader
R4	The use case should complete as an end-to-end transaction (which can include alternative/exceptional flows) with appropriate post-conditions satisfied	Structure building framework; adjacency pair principle		7 C's Facet 2 (Phalp et al. 2007b), Cockburn (1995): completion of responsibilities	All types of reader

Table 2 continued

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R5	The logic of the use case description should provide a rational answer to the problem	Pragmatic principle number 4: make true claims about the situation model; pragmatic principle number 7: statements should not contradict one another; structure building framework; importance of avoiding disrupting inference generation		7 C's Facet 2 (Phalp et al. 2007b); Anda et al. (2001)	All types of reader
R6	When an action occurs, there should be a meaningful response to that action	Event-Indexing Model: important to avoid causal discontinuity; structure building framework; adjacency pair principle; relates to causal relationships and coherence	Assists use case in following a logical path	CP Style Rule 8 (Cox et al. 2001); 7 C's Facet 2 (Phalp et al. 2007b); Schneider and Winters (2001)	All types of reader
R7	Show the actor's intent, not the actor's movements in operating the system's user interface		Cockburn (2001) believes that describing the user's movements during the operation of the system's user interface is "one of the more common and severe mistakes"	Cockburn's Guideline 5 (2001)	All types of reader

Table 2 continued

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R8	The use case should be at a consistent level of abstraction throughout		Mixing abstraction levels may make the use case more difficult to understand (Phalp and Cox 2003)	7 C's Facet 4 (Phalp et al. 2007b)	This rule is not appropriate for use cases intended for designers who may think and move through different levels of abstraction when considering a design problem
R9	Each sentence in the description should be on a new, numbered line		Clarity of narrative; ease of identification; increased traceability	CP Style Rule 1 (Cox et al. 2001); CREWS SG1 (Ben Achour et al. 1999); Anda et al.'s (2001) style guidelines SG1; Cockburn (2001), Kulak and Guiney (2000), Schneider and Winters (2001) (optional)	All types of reader
R10	Alternatives and exceptions should be described in a section below the main description, and the sentence numbers should agree	Pragmatic Principle no. 6: order of mentioning events should correspond to the chronological order of events in the situation model; pragmatic principle no. 7: statements should not contradict one another	Main flow is easy to read; plenty of room to provide details of alternatives; no need to change numbering if alternative paths added later	CP Style Rule 1 (Cox et al. 2001); 7 C's Facet 7 (Phalp et al. 2007b); CREWS SG2 (variations); CREWS SG3 (alternatives) (Ben Achour et al. 1999); Anda et al.'s (2001) style guidelines SG2 (variations); Anda et al.'s (2001) style guidelines SG3 (alternatives); Cockburn (2001), Kulak and Guiney (2000), Schneider and Winters (2001) (optional)	All types of reader

Table 2 continued

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R11	Use simple present tense throughout	Complex structures increase reading time and are sometimes misinterpreted	This may be particularly appropriate for multinational projects	CP Style Rule 6 (Cox et al. 2001); 7 C's Facet 6 (Phalp et al. 2007b); CREWS SG5 (Ben Achour et al. 1999); Anda et al.'s (2001) style guidelines SG5 (present tense); Cockburn (2001); Kulak and Guiney (2000) (simple); Schneider and Winters (2001) (simple)	All types of reader
R12	Use the active not the passive voice	The passive voice is more difficult to understand and takes more time to read and formulate than its active equivalent; the "by" phrase which contains the agent in a passive sentence may lack prominence	The passive voice may encourage the agent of the verb to be omitted	implied by CP Style Rules (Cox et al. 2001); CREWS SG5 (Ben Achour et al. 1999); Anda et al.'s (2001) style guidelines SG5; Cockburn (2001), Kulak and Guiney (2000)	All types of reader
R13	Avoid the use of structures which resemble programming language, e.g. "if-then-else" statements and "end loop"		Use cases assist communication between domain experts, end-users and developers: they should be written in the user's vocabulary; implementation-specific language should be avoided; such structures break the flow of the use case narrative	Cockburn (2001), Kulak and Guiney (2000)	This rule is not necessary for use cases which are intended for technical readers, e.g. designers, programmers

Table 2 continued

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R14	Each event should be written in the form "Subject… verb… object… (optional:) prepositional phrase"	NVN strategy is very influential; complex structures increase reading time and are sometimes misinterpreted	In a survey of over 150 use cases, "subject verb object" was the most frequently used sentence structure (Cox et al. 2001)	CP structure rules 1 and 2 (Cox et al. 2001); CREWS CG5 and 1-3 (Ben Achour et al. 1999); Anda et al.'s (2001) Style Guidelines CG2; Cockburn (2001)	The "subject verb object" structure is, in itself, highly suitable for all types of readers. This rule as it stands may not be strictly appropriate for use cases intended for readers with poor domain knowledge, e.g. designers and programmers
R15	Avoid pronouns, synonyms and homonyms	Readers may sometimes fail to identify a unique referent for a pronoun; Gernsbacher's explicitness principle reduces importance of pronouns; suitability of pronouns depends on their context, therefore care needed in their positioning	Reduced risk of ambiguity and therefore errors	CP Style Rule 2 (pronouns, at least partially) (Cox et al. 2001); 7 C's Facet 6 (pronouns and synonyms) (Phalp et al. 2007b); CREWS SG4 (pronouns, synonyms and homonyms) (Ben Achour et al. 1999); Anda et al.'s (2001) style guidelines SG4 (synonyms and homonyms)	All types of reader
R16	Avoid adverbs and adjectives	Complex structures increase reading time and are sometimes misinterpreted Designers may benefit from identifying adjectives as services within objects	Adverbs and adjectives introduce subjectivity which can encourage misunderstanding	CP Style Rule 3 (Cox et al. 2001); 7 C's Facet 6 (Phalp et al. 2007b); CREWS SG6 (adverbs only) (Ben Achour et al. 1999); Anda et al.'s Style (2001) Guidelines SG6 (adverbs only)	This rule is not suitable for use cases intended for readers with low domain knowledge

Table 2 continued

	Rule	Supporting discourse process theories	Other supporting arguments	Use case guidelines and guidance	Intended readership
R17	Avoid negatives except in alternative and exceptional flows	Complex structures increase reading time and are sometimes misinterpreted	Main flow should describe shortest successful path; therefore, it should not require negatives; paths handing alternatives, exceptions and failures may require negatives	CP Style Rule 4 (avoid negatives in general) (Cox et al. 2001); 7 C's Facet 6 (avoid negatives in general) (Phalp et al. 2007b); CREWS SG6 (Ben Achour et al. 1999); Anda et al.'s (2001) Style Guidelines SG6 (avoid negatives in general)	All types of reader
R18	Underline other use case names referenced in a use case		This increases the ease with which referenced use cases can be identified; therefore, it increases clarity; underlining is preferable to the use of italics; tools may allow a hyperlink from the event to the referenced use case	CP Structure Rule 3 (Cox et al. 2001); Cockburn (2001)	All types of reader

References

Adlem, A. (2007). *A critique of use case guidelines*. Thesis Bournemouth.

Anda, B., Sjoberg, D. I. K., & Jorgensen, M. (2001). Quality and understandability of use case models. *Lecture Notes in Computer Science, C2001*, 402–428.

Bamberg, M., & Moissinac, L. (2003). Discourse development. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes*. Mahwah, NJ: Lawrence Erlbaum Associates.

Ben Achour, C., Rolland, C., Maiden, N. A. M., & Souveyet, C. (1999). Guiding use case authoring: Results of an empirical study. *4th IEEE international symposium on requirements engineering (RE'99)*, June 7–11, 1999. Ireland: University of Limerick. http://sunsite.informatik.rwth-aachen.de/CREWS/reports.htm.

Bray, I. K. (2002). *An introduction to requirements engineering*. Harlow: Addison-Wesley.

Britton, B. K., & Gulgoz, S. (1991). Using Kintsch's computational model to improve instructional text: Effects of repairing inference calls on recall and cognitive structures. *Journal of Educational Psychology, 83*(3), 329–345.

Cockburn, A. (1995). Structuring use cases with goals. Resource document. *Journal of Object Oriented Programming, 1997*. http://alistair.cockburn.us/index.php/Structuring_use_cases_with_goals. Accessed 1 Feb 2009.

Cockburn, A. (2001). *Writing effective use cases*. London: Addison-Wesley.

Cox, K., Aurum, A., & Jeffery, R. (2004). An experiment in inspecting the quality of use case descriptions. *Journal of Research and Practice in Information Technology, 36*(4), 211–229.

Cox, K., Phalp, K., & Shepperd, M. (2001). Comparing use case writing guidelines. *7th international workshop on requirements engineering: foundation for software quality*. June 4–5, 2001, Interlaken, Switzerland: Essen, Essener Informatik Beitrage.

Crystal, D. (1997). *The Cambridge encyclopaedia of language*. Cambridge: Cambridge University Press.

Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM, 31*(11), 1268–1287.

Curtis, B., Walz, D., & Elam, J. (1989). Studying the process of software design teams. *5th international software process workshop*. Kennebunkport, Maine, USA: IEEE Computer Society Press.

Ferreira, F. (1994). Choice of passive voice is affected by verb type and animacy. *Journal of Memory and Language, 33*(6), 715–736.

Ferreira, F. (2003). The misinterpretation of noncanonical sentences. *Cognitive Psychology, 47*, 164–203.

Fletcher, C. R., Van Den Broek, P., & Arthur, E. J. (1996). A model of narrative comprehension and recall. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 141–163). Mahwah, NJ: Lawrence Erlbaum Associates.

Foltz, P. W. (2003). Quantitative cognitive models of text and discourse processing. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes* (pp. 289–311). Mahwah, NJ: Lawrence Erlbaum Associates.

Gernsbacher, M. A. (1996). The structure-building framework: What it is, what it might also be, and why. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 289–311). Mahwah, NJ: Lawrence Erlbaum Associates.

Gibbs, R. W. (1996). Metaphor as a constraint on text understanding. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 215–240). Mahwah, NJ: Lawrence Erlbaum Associates.

Glass, R. (1998). *Software runaways*. Harlow: Prentice Hall.

Goldman, S. R., Varma, S., & Cote, N. (1996). Extending capacity-constrained construction integration: Toward "smarter" and flexible models of text comprehension. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text*. Mahwah, NJ: Lawrence Erlbaum Associates.

Gordon, P. C., & Chan, D. (1995). Pronouns, passives, and discourse coherence. *Journal of Memory and Language, 34*, 216–231.

Gordon, P. C., Grosz, B. J., & Gilliom, L. A. (1993). Pronouns, names, and the centering of attention in discourse. *Cognitive Science, 17*, 311–347.

Gordon, P. C., & Scearce, K. A. (1995). Pronominalization and discourse coherence, discourse structure and pronoun interpretation. *Memory and Cognition, 23*(3), 313–323.

Graesser, A. C., Gernsbacher, M. A., & Goldman, S. R. (2003). Introduction to the handbook of discourse processes. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes*. Mahwah, NJ: Lawrence Erlbaum Associates.

Graesser, A. C., Millis, K. K., & Zwaan, R. A. (1997). Discourse comprehension. *Annual Review of Psychology, 48*, 163–189.

Graesser, A. C., Singer, M., & Trabasso, T. (1994). Constructing inferences during narrative text comprehension. *Psychological Review, 101*(3), 371–395.

Greene, S. B., McKoon, G., & Ratcliff, R. (1992). Pronoun resolution and discourse models. *Journal of Experimental Psychology. Learning, Memory, and Cognition, 18*(2), 266–283.

Harley, T. A. (2001). *The psychology of language: From data to theory* (2nd ed.). Hove: Psychology Press.

Johnson-Laird, P. N. (1968). The choice of the passive voice in a communicative task. *British Journal of Psychology, 59*, 7–15.

Kintsch, W., & Van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review, 85*, 363–394.

Kulak, D., & Guiney, E. (2000). *Use cases: Requirements in context*. London: ACM Press.

Mannes, S. M., & Kintsch, W. (1987). Knowledge organization and text organization. *Cognition and Instruction, 4*(2), 91–115.

Mannes, S. M., & St. George, M. (1996). Effects of prior knowledge on text comprehension: A simple modeling approach. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 115–139). Mahwah, NJ: Lawrence Erlbaum Associates.

McKoon, G., Gerrig, R. J., & Greene, S. B. (1996). Pronoun resolution without pronouns: Some consequences of memory-based text processing. *Journal of Experimental Psychology. Learning, Memory, and Cognition, 22*(4), 919–932.

McNamara, D. S., Kintsch, E., Songer, N. B., & Kintsch, W. (1996). Are good texts always better? Interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction, 14*(1), 1–43.

Moore, J. D., & Weimer-Hastings, P. (2003). Discourse in computational linguistics and artificial intelligence. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes* (pp. 439–485). Mahwah, NJ: Lawrence Erlbaum Associates.

Phalp, K., & Cox, K. (2002). Supporting communicability with use case guidelines: An empirical study. In *6th international conference on empirical assessment in software engineering*. Keele: Keele University.

Phalp, K. T., & Cox, K. (2003). Exploiting use case descriptions for specification and design. In *7th international conference on empirical assessment and evaluation in software engineering (EASE 2003)*, April 8–10th, 2003. Staffordshire, UK: Keele University.

Phalp, K., Jeary, S., & Adlem, A. (2010). *The application of discourse process theory to use case descriptions*. Bournemouth: Bournemouth University. TR-SSRC01-2010.

Phalp, K., Vincent, J., & Cox, K. (2007a). Improving the quality of use case descriptions: empirical assessment of writing guidelines. *Software Quality Journal, 15*(4).

Phalp, K. T., Vincent, J., & Cox, K. (2007b). Assessing the quality of use case descriptions. *Software Quality Journal, 15*(1), 69–97.

Rolland, C., & Ben Achour, C. (1998). Guiding the construction of textual use case specifications. *Data & Knowledge Engineering, 25*, 125–160.

Schneider, G., & Winters, J. P. (2001). *Applying use cases: A practical guide* (2nd ed.). London: Addison-Wesley.

Stanfield, R. A., & Zwaan, R. A. (2001). The effect of implied orientation derived from verbal context on picture recognition. *Psychological Science, 12*(2), 153–156.

Van Den Broek, P., Risden, K., Fletcher, C. R., & Thurlow, R. (1996). A “landscape” view of reading: Fluctuating patterns of activation and the construction of a stable memory representation. In B. K. Britton & A. C. Graesser (Eds.), *Models of understanding text* (pp. 165–187). Mahwah, NJ: Lawrence Erlbaum Associates.

Zwaan, R. A., & Singer, M. (2003). Text comprehension. In A. C. Graesser, M. A. Gernsbacher, & S. R. Goldman (Eds.), *Handbook of discourse processes* (pp. 83–121). Mahwah, NJ: Lawrence Erlbaum Associates.

Zwaan, R. A., Stanfield, R. A., & Yaxley, R. H. (2002). Language comprehenders mentally represent the shapes of objects. *Psychological Science, 13*(2), 168–171.

Author Biographies

Keith Phalp originally read for a first degree in Mathematics, which he then taught for a few years, before completing a Masters in Software Engineering, followed by Ph.D. in Process Modelling. He then spent 3 years as a post-doctoral research fellow at the University of Southampton, again in the area of process modelling. In 1997, Dr. Phalp took up a lectureship at Bournemouth University and has been there ever since in a variety of roles. He is currently an Associate Dean and the Head of Academic Group for Software Systems and Psychology.

Anita Adlem has a background in English and came to Software Engineering as a career change. However, she found that her affinity for the study of language led her to question its use within requirements and, hence, to undertake work in this area as post-graduate student.

Sheridan Jeary is Co-Director of the Software Systems Research Centre at Bournemouth University and has a Ph.D. in Web Development Methods. Her research interests span Web Systems and the alignment of Information Technology with business (strategy and process). She had several years of management and systems experience across a variety of domains before moving into academia. She was the BU Project Manager for the EU Commission funded VIDE project on Model-driven development and is currently investigating the production of useful models in complex enterprises.

Jonathan Vincent has a B.E. in Electrical and Electronic Engineering, an M.Sc. in Computing (Software Engineering) and a Ph.D. in Computer Science. He has wide-ranging research interests and has published in a variety of areas within software engineering and computer science, including component-based software engineering, software quality, modelling, evolutionary computation and neural networks.

John Kanyaru has a B.Sc. in Computer Science and a Ph.D. in software engineering. His research interests are within Requirements Engineering, in particular, the use of prototyping to support validation of specifications. Other emerging research interests are within model-driven software development and medical informatics.