Instructional Science 21: 183–198 (1992)

Reviewing and Correcting Specifications

ANTHONY FINKELSTEIN

Imperial College of Science, Technology and Medicine,
Department of Computing, 180 Queen's Gate, London SW7 2BZ.
Email: acwf@doc.ic.ac.uk

Abstract. We outline a scheme for marking suggested edits and annotations on software specifications, a particularly complex class of structured document, during the process of review and correction. The scheme is based on a formal model of document construction and review and on typographic marking methods. The scheme permits precise and interpretable marking and annotation of documents which use many different notations. It supports and guides the process of correction. Some examples and a sample visual notation are given. Tool support for using this scheme is briefly discussed.

INTRODUCTION

Specifications, like all large and complex documents, are prone to error, whether simply slips or more serious faults. The process of eliminating errors has two parts, spotting an error and correcting the specification. In the study of specification much attention has been paid to the business of spotting errors through, for example, animation, automated reasoning and inspection, but virtually no attention has been paid to that of correcting specifications.

The task of correcting a specification is, at least, as complex as constructing a specification de novo. Once the error has been spotted, and tentatively tied down to a part (or parts) of the specification, a strategy for remedying the error must be developed and agreed. It must be ascertained whether the error is, indeed, an error and whether the cost of correcting it exceeds the likely costs incurred by allowing it to remain in the specification. Where correcting the specification requires expertise or authority outside that of the person who spotted the error this needs to be obtained. This may mean informing further people of the nature and location of the error.

Specifiers are seldom very disciplined about this process. Specifications are distributed, in draft form, to reviewers who, having applied their tools and analytical skills, will return comments, questions and corrections. Unless a special purpose comment form or questionnaire has been included the reviewer will most often scribble on the draft itself, occasionally complementing the draft with 'post-it' notes or memos.

It is, in principle, possible for a reviewer using automated tools to directly edit the specification creating a new version which can subsequently be merged into the finalised specification. However, many

of the reviewers comments are not direct edits but rather requests for clarification and the like. Even where there is a need to make direct edits it is often too difficult for the reviewer to learn to use the variety of high functionality automated tools that may be necessary.

This gives rise to a need for systematic techniques for annotation and marking suggested edits and comments on specifications, both manual and computer-based. These annotations need to be both precise and easy to interpret. Below, we outline a scheme for marking these on specifications. We examine the typographic and copy-editing approach to document correction and identify its strengths and shortcomings. We then outline a formal model of how commitments are negotiated and established and we use this to underpin a marking scheme. We give a sample visual notation for this scheme and illustrate how it is used for some small examples. We discuss the relation between our approach and other methods and give a short account of tool support and scaling-up.

SPECIFICATIONS

Software specifications are a particularly complex class of structured document. They are subject to considerable review and correction. Errors introduced at the specification stage in software development become increasingly costly to correct as that development proceeds, hence it makes sense to fix errors as early as possible

Software specifications consist of many different types of knowledge couched in a mixture of representation schemes — formal, informal and formatted. Most specifications are tapestries of different representation schemes. Errors and consequently the need to correct the specification span across many representation schemes. Such specifications are reviewed by many different people with different and partial knowledge of the representation schemes used. The specification and any subsequent 'fixes' are the result of a collaboration between many people each of whom has a stake in it. In the discussion which follows we examine how we can support the process of review and correcting errors in such documents.

DOCUMENT PREPARATION METHODS

Over time publishers and others who have to produce many complex documents have developed schemes for removing as many errors as possible from their documents. These schemes have, for obvious reasons, concentrated on correcting documents rather than on spotting errors.

We may divide the processes of preparation into three broad phases, acquisition, copy-editing and proof. In this section we will largely be concerned with copy-editing.

Copy-editing begins once the manuscript has been placed before the copy-editing department by the acquisitions editor. The copy-editor will mark the edits on the manuscript. Different colour pencils are used to distinguish who is marking what — there are accepted conventions governing the colours used.

The edits are marked using a simple scheme which allows the author to examine the edits and determine whether they are correct and appropriate. Each editing operation has an associated pair of marks — a text mark, to appear in the text at the exact place where the edit is to be made and a marginal mark to signify and amplify the meaning of the text mark. There are special marks for types of error which frequently occur.

Tables 1 and 2 show a small sample of copy-editing marks taken from BS 5261, the UK standard on copy preparation and proof correction. Some are probably familiar, others less so. Certain marks can be combined to give composite edits, G and H to give I. Some marks are more complex combinations. Characters or words to be changed to capitals are indicated by double underlining and substitution is indicated by a single angled stroke through characters to be changed. These can be combined to give marginal mark J. The combination of marks is not consistent, thus L is the reverse of K but the marks are not related. Some marks have alternatives which can be used when the preferred mark is difficult to use. Thus N is used when the preferred mark M does not allow the sequence to be seen clearly.

Marginal marks generally appear on the margin closest to the text mark and are sequenced so that the farthest text mark from the edge of the block of text corresponds to the margin mark nearest the edge of the block of text. Each mark is concluded with a single angled stroke. For an illustration of this see Figure 1.

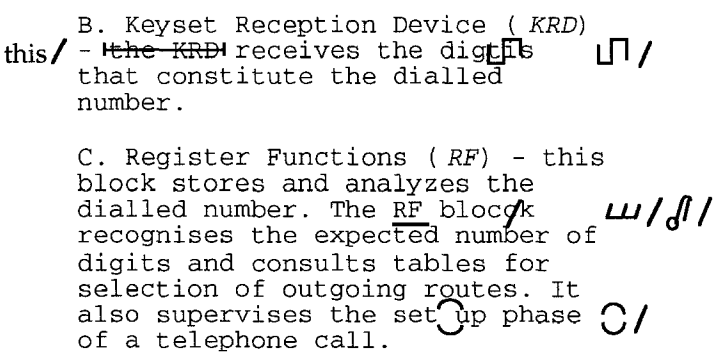


Figure 1. Illustration of the use of copy editing marks

Instruction	Text mark	Margin mark	Note
Leave unchanged	*under characters to remain*		A
Insert in text the matter indicated in the margin			B
Insert in text the matter indicated in the margin		*followed by for example* A	The relevant section of copy should be supplied with the corresponding letter marked on it in a diamond C
Delete	*through characters* or *through words*		D
Delete and close up	*through characters* or *through words*		E
Substitute	*through characters* or *through words*	*new character* or *new word(s)*	F
Change to italic	*under characters to be changed*		G

Table 1. Copy-editing marks from BS 5261

Instruction	Text mark	Margin mark	Note
Change to bold	*under characters to be changed* 		H
Change to bold italic	*under characters to be changed* 		I
Change capital letters to lower case letters	*circle characters to be changed*		J
Start new paragraph			K
Run on			L
Transpose characters or words	*between characters or words* 		M
Transpose characters or words		123	The vertical strokes are made through characters or words to be transposed N

Table 2. Copy editing marks, continued.

An important feature of copy-editing marks is that they are (more or less) independent of the language in which the document is written. The same marks apply if the document is in Italian or English (clearly, certain languages such as Chinese may be more problematic). Special additional copy-editing marks are used for mathematical notation. Once the copy-editor has completed marking up the document it is returned to the author, generally with a covering letter giving a rationale for the changes. The author will check through the edits to ensure that they are correct and appropriate. The document may go through several copy-edit — author correction cycles before the manuscript is accepted by both parties as complete.

The completed manuscript is sent to the printer who will prepare a proof or early printed version which shows type, justification, and so on, as they will appear in the finished document. Some specialist marks to denote common printers errors such as marks from visible paper edges are used.

Increasingly automated approaches are being adopted to marking formatting during document preparation. These are likely to make much of the traditional style editing task obsolete. Nevertheless the conventional techniques still apply in syntactic and substantive editing or in documents where limited and highly localised style changes are to be made to a pre-existing text.

ANALYSIS

The use of typographic and copy editing marks in traditional document preparation suggests the possibility of applying them to review and correction of specifications.

These marks have the following benefits:

- they provide a categorisation of common types of edits and queries and thus provide a vehicle for communication between reviewer and author;
- they are simple to understand and manipulate;
- they are independent of the underlying representation and hence can be used in documents consisting of heterogeneous representations;
- they mark errors and queries explicitly and by identifying potential conflicts support the resolution of disagreements.

The existing copy-editing and proof correction marks are, however, of limited immediate use in specification review, annotation and correction being very low-level, in the sense of largely addressing character and word manipulation rather than larger text units, and concentrating on syntactic rather than semantic elements of the document.

Below we attempt to develop a scheme of marks and give a sample notation which:

- preserves and builds on the benefits of copy-editing and proof correction marks;
- addresses corrections and annotations to specifications at the appropriate level;
- is (notationally) consistent and well (formally) defined;
- is amenable to automated support.

MODEL OF SPECIFICATION CONSTRUCTION AND REVIEW

To develop a scheme with the properties outlined above we require a model of document construction and review to underpin it. We have developed such a model for specifications which is based on how

'commitments' are negotiated and established as a specification is constructed. The model, which develops concepts taken from dialogue and commitment logics, is described in full in Finkelstein and Fuks (1989). Below we give a brief summary of its main concepts and features and go on in subsequent sections to show how it might be applied.

A specification, at its most basic level, consists of a set of statements. These statements may be textual, graphical, mathematical or some hybrid. A simple example is:

"All transactions with the auto-teller are in units of £5"

Each such statement is an effective restriction on the freedom of action on the part of the developer who must build a system that satisfies the specification. By making a statement the specifier is, in effect, making a 'commitment' that is, holding him or herself out as liable for the consequences of that statement. If, in the example above, the specifier subsequently demands that the auto-teller allows customers to withdraw cash in any units they demand, we would not be surprised if the developer complained.

We see specification as a process by which commitments are negotiated and established by the parties having responsibility for the description and development of the system. This process takes the form of a dialogue or organised sequence of locutions.

Our model permits the following locution types:

ask a question "Is it the case that...?";

make an assertion "It is the case that...";

make a denial "I deny that it is the case that...";

withdraw a statement "No commitment to...";

make a challenge "Why is it to be supposed that...";

demand that the consistency of the other participant's commitments is checked (resolve commitments) "Are these commitments consistent?";

demand that the consistency of the other participant's commitments is checked with respect to a given statement (resolve consequences) "Given a commitment to...are your commitments consistent".

Dialogues have an 'etiquette' which governs the legitimate shape of the interaction. The permitted sequences are described in dialogue (or response) rules. Performing a locution will generally have an effect on the commitments of both the 'speaker' and the 'hearer'. In our model we give commitment rules that define these effects. Additionally we can define, syntactically, the form of reasoning permissible within the dialogue and common to its participants. Our model primarily involves

'modus ponens', though addition of other schemas to fit various logical tastes is a relatively simple matter.

In the section which follows we outline how this model can be applied to support a scheme for review, annotation and correction.

SCHEME

In this section we develop a scheme for review, annotation and correction of specifications consequent upon the discovery of a flaw in the specification, generally during validation. The scheme has a formal underpinning described in the model above and is inspired by and develops the visual marking schemes used in traditional document preparation.

We use a simple graphic notation for our marking scheme which, with minor variations, can be applied to specifications in natural language and formal notations or some mixture of both. We are not particularly attached to our notation, and it is intended as illustrative.

As a starting point it is, of course, important to distinguish who is reviewing, annotating and correcting what. We shall consider two participants which, following the copy editing convention we shall distinguish by the colour used for marks: black and blue. In our figures blue is shown as shaded. In the following discussion we distinguish two roles specifier and reviewer. These roles are interchangeable thus if initially black is the specifier, once black has handed over the specification for comment to blue and blue starts making marks on the specification the roles are reversed: black is the reviewer and blue the specifier. Black always starts. This shifting of roles can be confusing and should be kept in mind when reading the description and example which follows.

For each of the locution types outlined above there is an equivalent in our scheme, see Table 3. We have employed the graphic principles underlying existing mark schemes. A question (mark i), is indicated by a line underneath the questioned statement. The filled triangle indicates its start and the blank triangle its finish. The margin mark appears closest to the beginning of the marked text with the conventional stroke to indicate conclusion.

Figure 2 illustrates how this mark is used. In this example the marker, uncertain of the truth of the statement that the B-subscriber is a member of the same exchange, marks it as in question, in effect asking the the other participant "Is it the case that the B-subscriber is a member of the same exchange?".

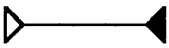
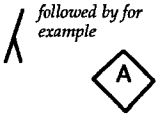
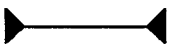
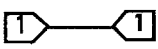
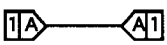
Instruction	Text mark	Margin mark	Note
Question	*under text to delimit commitment* 		i
Denial	*under text to delimit commitment* 		ii
Assert		 *followed by for example*	Insert the statement indicated in the margin. The relevant statement should be supplied with the corresponding letter marked on it in a diamond iii
Withdrawal	*under text to delimit commitment* 		iv
Challenge	*under text to delimit commitment* 		v
Resolve Commitments	*under text to delimit commitments to be resolved* 		vi
Resolve Consequences			Check consistency with the statement indicated in the marginand inthe text mark. The relevant statement should be supplied with the corresponding letter marked on it in a diamond vii

Table 3. Scheme for review, annotation and correction of specifications.

The last digit is received by RF which analyses the dialled number and recognises that the B-subscriber, which is a member of the same exchange, is the recipient of the call. Then RF sends the SELSUB signal to request the selection of that subscriber from the appropriate LIC. The B-subscriber is linked to this LIC, denoted LIC(B).

Figure 2. Illustration of the marking scheme.

Figure 3 gives a small fragment of text marked up as if reviewed. There are some simple syntactic rules governing the application of these marks and, as in our dialogue model, the way in which a text has been marked conditions the responses that can be made to it. We can state these in dialogue rules that describe the ways in which replies may be made. Thus, for example, the reply to a challenged statement must be the withdrawal of the statement or it must be the assertion of a statement to which the challenger is not committed. Similarly each mark has an effect on the commitments of the participant making that mark, the specifier, and on the commitments of the participant who receives the marked document, the reviewer. These are given in the form of commitment rules. For example, assertions add a commitment on the part of the specifier to the assertion and also have the effect of committing the reviewer.

For a full account of these rules and their operation the reader should refer to Finkelstein (1990).

USING THE SCHEME

In this section we consider in more detail how the review, annotation and correction scheme discussed above can be used. Let us consider a typical scenario. The specifier (who only has limited knowledge about the domain) will not be fully aware of the consequences of some of the corrections that he or she is making. That is, the consequences of marks when placed in the context of unmarked text which have become part of the commitments by 'default'.

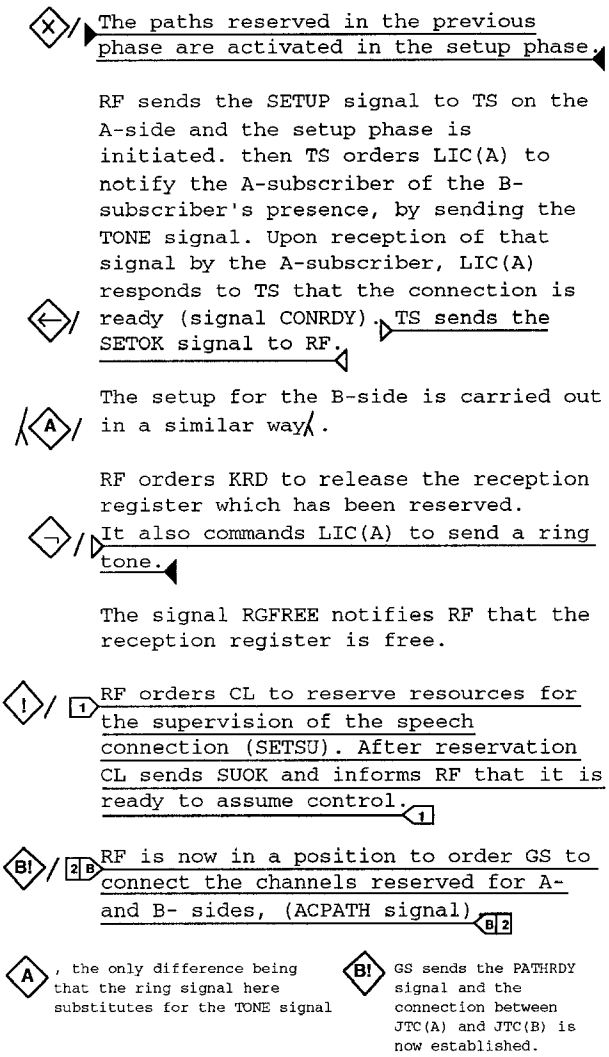


Figure 3. A fragment of text marked up as if reviewed.

Typically the reviewer's response will be to demand that the specifier checks for inconsistencies in some marked portion of what are now the specifier's commitments or determines their consistency against some given statement. Alternatively, the reviewer may simply deny a statement asserted by the specifier. In such a manner corrections bounce back and forward until there are no further outstanding marks and all changes to the specification have been agreed. There may of course come a point where one side or another will have to exert authority to ensure that this is the case. A simple fixing sequence is shown in Figures 4 and 6. In the columns on the left and right hand sides of the figures we indicate the specifying party, the position in the sequence of marks and, for pedagogic reasons, an explanatory comment.

Because we wish to avoid the situation where the specification becomes cluttered with marks we assimilate the changes into the main body of the specification as we go along. This leads to the following three step process: the specifier marks the document; the marked specification is inspected by the reviewer who then adopts the role of specifier and marks on top of the existing marks (which are in another colour); the changes made in the preceding set of changes are assimilated into the specification and the cleaned up specification is passed on. The assimilation of changes is done in accordance with the commitment rules.

For example in Figure 4 step [2] blue makes an assertion. Black is given the marked copy and the roles, specifier and reviewer, are exchanged. Black then makes changes on top of the marked text, this intermediate step is shown in Figure 5, and the blue changes are assimilated into the text as shown in Figure 6 step [3].

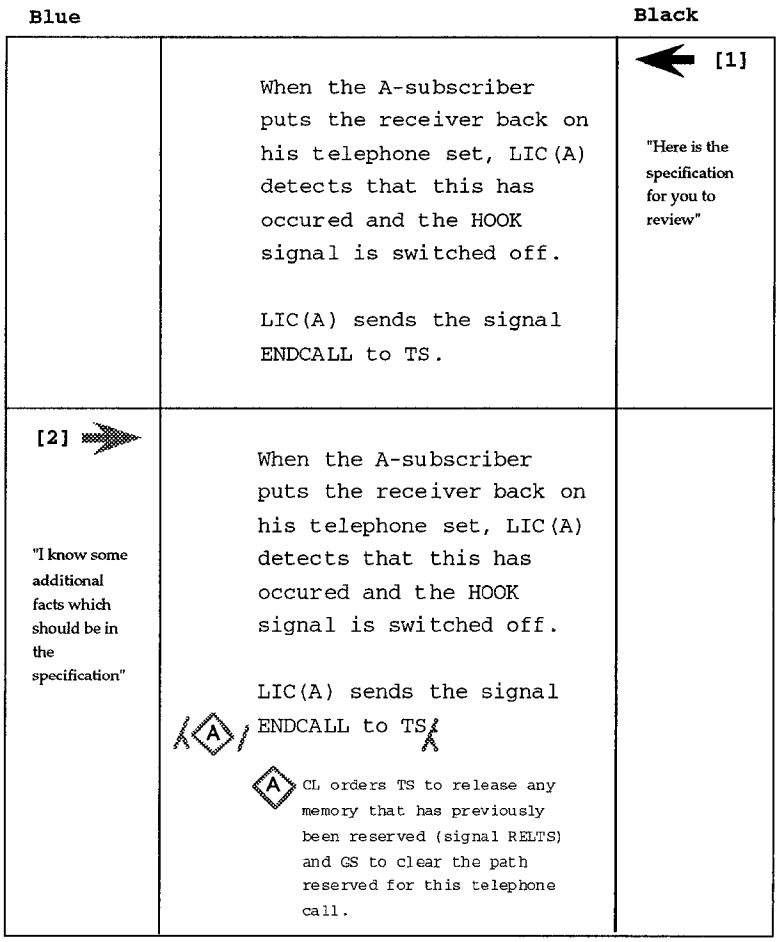


Figure 4. Fixing sequence, steps 1 and 2.

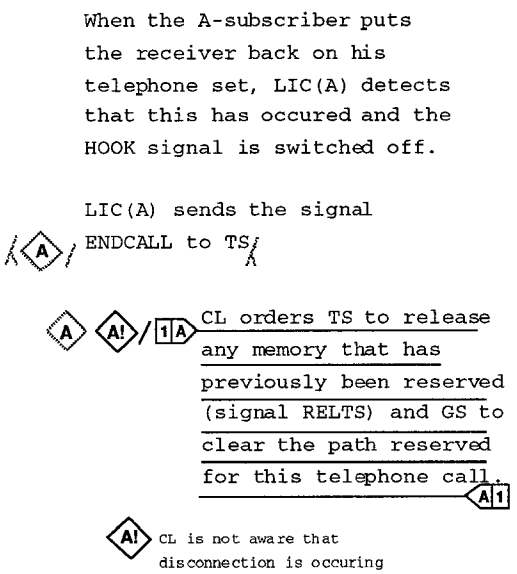


Figure 5. Fixing sequence, intermediate step.

EXPERIENCE AND TOOL SUPPORT

We believe that our scheme is of value to discipline and make systematic the process of review, annotation and correction. We have gained experience on a significant number of small examples and are currently working on larger examples from the telecommunications and computer integrated manufacture domain. Clearly our scheme is amenable to automated support. We are developing a simple 'desk accessory' which can be used alongside existing CASE and document preparation tools and will support the use of the scheme described above. To this end we have simulations of the underlying model both in Prolog and Smalltalk and are working a hypertext based tool which will allow us to examine the application of the scheme in more detail. This tool will deploy a revised notation using highlighting and icons.

We have not, to date, considered in detail the complexities that can arise when many people concurrently revise a specification though they are clearly important. This topic will be receiving further attention.

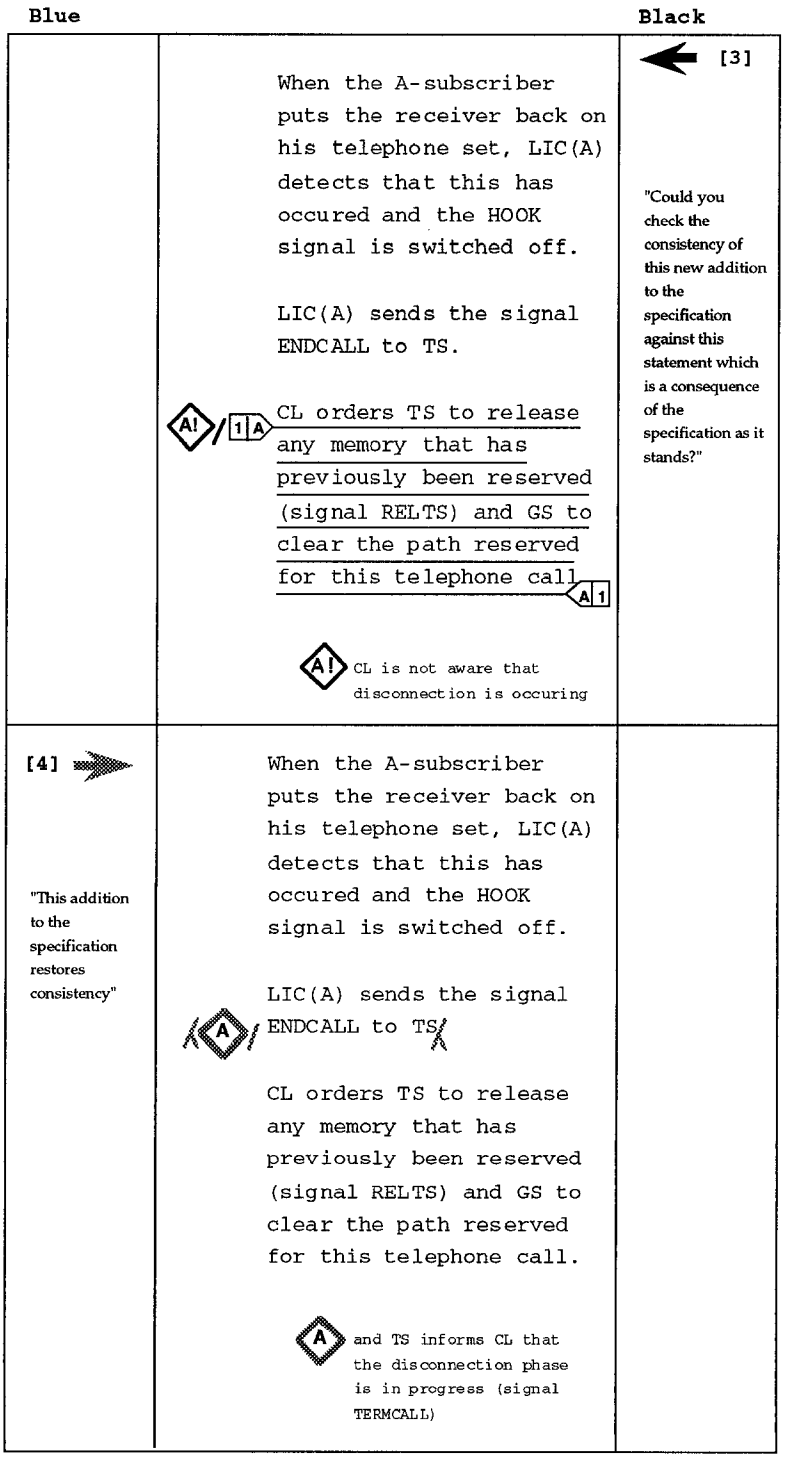


Figure 6. Fixing sequence, steps 3 and 4.

COMPARISON AND RELATIONS

We have been unable to find any precise equivalent to the approach we have adopted here. There are however a number of related areas which are worthy of mention.

For details of traditional approaches to document preparation reference may be made to The University of Chicago (1982) or to Butcher (1975). Some interesting comments on hand marking of documents can be found in Gould and Salaun (1987) also Welbourn and Whitrow (1988). Hartley (1984) has examined the comments provided by colleagues about an academic article. Many of the categories of comment he observes are supported within our scheme. His work suggests that some empirical analysis of specification marking and editing might be worthwhile.

In the area of software engineering Fickas and Nagarajan (1988) have conducted some interesting work on critiquing specifications. What we provide can be seen as complementary to this work. Feather (1987) examines how specifications are constructed, his use of the term high-level editing is suggestive of links with the work outlined above.

Clearly our approach, in which we emphasise the cooperative nature of specification relates to concerns shared by workers in the area of computer supported cooperative work. Particularly relevant is work on cooperative writing. Sharples and Pemberton (1990) have reviewed work in this area, they provide a rough categorisation of meta-text objects (essentially those document items not intended for inclusion in a finished document). It is interesting to note that correction marks are the only class of such objects which have not received attention from those involved in specification.

Attaching annotations to software and specification objects is widely used (Reiss 1990). We have used notes attached to specifications as a basis for method guidance in a CASE tool (Kramer, Finkelstein et al., 1987). An interesting development in this area is the use of hypertext to support software development notably Conklin (1989).

CONCLUSIONS

What have we achieved with this scheme? We have developed a simple way of reviewing, annotating and supporting the correction of specifications. It gives a discipline to the ill organised activity of review, annotation and correction specifications by providing guidance and support to both specifier and reviewer. With a fairly simple set of marks rich and complex annotations can be made. The scheme is precisely interpretable and can be applied to specifications consisting of many different representation schemes.

ACKNOWLEDGEMENTS

Thanks to my colleagues and students. The examples are extracted from Votsis and Papoulias (1989).

REFERENCES

Butcher, J. (1975) *Copy-editing: the Cambridge Handbook*; CUP, Cambridge.

Conklin, J. (1989) 'Design Rationale and Maintainability' *Proc. 22nd Hawaii International Conference on System Sciences*; II, pp. 533-539; IEEE CS Press.

Feather, M. (1987) 'Constructing Specifications by Combining Parallel Elaborations' *IEEE Trans. Software Engineering.*

Fickas, S. and Nagarajan, P. (1988) 'Being Suspicious: Critiquing Problem Specifications' *Proc AAAI 88*; **1**, pp. 19-24; Morgan Kaufmann.

Finkelstein, A. (1990) 'A Scheme for Review, Annotation and Correction of Specifications' *Imperial College, Department of Computing, Technical Report 90.*

Finkelstein, A. and Fuks H. (1989) 'Multi-Party Specification' *Proc 5th International Workshop on Software Specification and Design*, pp 185–195; IEEE CS Press (also as Special Issue of ACM Software Engineering Notes).

Gould, J. and Salaun, J. (1987) 'Behavioural Experiments on Handmarkings' (full reference not available).

Hartley, J. (1984) 'The Role of Colleagues and Text-Editing Programs in Improving Text' *IEEE Trans. on Professional Communication*; PC27,1.

Kramer, J., Finkelstein, A., Ng, K., Potts, C. and Whitehead, K. (1987) 'Tool Assisted Requirements Analysis: TARA final report' *Imperial College, Dept. of Computing, Technical Report 87/18.*

Reiss, S. (1990) 'On the Use of Annotations for Integrating the Source in a Program Development Environment' in Finkelstein, A. Tauber, M. and Traunmuller R. (eds.) *Human Factors in Information Systems Analysis and Design*; North Holland, pp. 25-36.

Sharples, M. and Pemberton, L. (1990) 'Starting from the Writer: Guidelines for the Design of User-centred Document Processors', *Computer Assisted Language Learning,* **2,** 37–57.

The University of Chicago (1982) *The Chicago Manual of Style (13th Edition),* The University of Chicago Press, Chicago.

Votsis, G. and Papoulias, A. (1989) 'Internal Call Between Subscribers in the Same Digital Switching Exchange Equipment' *REX (Esprit Project 2080) Telecommunications Demonstrator.*

Welbourn, L. and Whitrow, R. (1988) 'A Gesture Based Text Editor' *HCI'88 – People and Computers IV,* Jones and Windsor (eds.), Cambridge University Press.