A Scheme for Review, Annotation and Correction of Specifications

Anthony Finkelstein

Imperial College of Science, Technology & Medicine,
Department of Computing
acwf@doc.ic.ac.uk

Abstract

In this paper we outline a scheme for marking suggested edits and annotations on specifications during the process of review and correction. The scheme is based on a formal model of negotiation and typographic marking methods used in conventional document preparation. The scheme permits precise and interpretable marking and annotation of complex structured 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 discussed.

Key Words: annotation; negotiation; review; requirements; specification.

1 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 specifications 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 remediating 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. It may be important to determine who is responsible for the error so as to eliminate further errors. 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. In particular we concentrate on requirements specifications, a particularly complex class of document subject to considerable review and correction. 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 an account of tool support and scaling-up.

2 Requirements Specifications

In this section we discuss requirements specifications, the most complex and least well understood documents in software development, and explain why they are of particular interest when investigating review annotation and correction.

We have concentrated on requirements specifications for two reasons. Errors introduced at the requirements specification stage become increasingly costly to correct as development proceeds, hence it makes sense to fix errors as early as possible. Reviewing and correcting requirements specifications is made significantly more complex, and hence more interesting, by their particular structure and content.

The commonly accepted definition of a requirements specification is that it is a document which establishes, as precisely as possible, the services that the software is required to provide and the constraints under which those services are provided. A systematic model of the real world from which these requirements arise provides a structure within which this information is presented. While this definition accounts for the central core of a requirements specification it is in fact a considerably more multi-faceted document than the definition suggests. It may contain: a large amount of general domain information to orient the developers; those design decisions fixed by policy; assessments of the feasibility of satisfying the requirements with available technologies; contractual information; an outline of the projected software development process; accounts of management procedures; quality assurance standards; and so on. As such it both addresses and is reviewed by a large and various audience with a mix of skills and interests.

A distinguishing feature of the requirements for large and complex systems is that knowledge about them is distributed among many people. Thus, for example,the instrument engineer understands the operation of the sensing component of a real-time process control system, the control engineer understands the type of control that is to be applied and the process engineer understands the material that is being processed. All three are essential to construct a requirements specification. Each presents information in different ways and is responsible for validating different aspects of the system. In addition there may be people who are responsible for "global" requirements such as performance. The system model itself and any subsequent "fixes" are the result of a collaboration between many people each of whom has a stake in the specification.

The means we have for expressing requirements are relatively poor. The representation schemes used can only express limited aspects of the requirements. The consequence of this is that no single representation scheme is a wholly appropriate medium in which to construct a requirements specification. For example, petri nets might be used to describe behaviour, entity-relationship-attribute diagrams to capture static data and data flow diagrams to express data transformation. The whole document might be glued together by passages of natural language which state non-functional requirements such as quantity and timing constraints, the relations which hold between the other parts of the specification, and so on. Most requirements specifications are thus tapestries of different representation schemes. Errors and consequently the need to correct the specification span across many representation schemes.

Thus, requirements specifications consist of many different types of knowledge couched in a mixture of representation schemes - formal, informal and formatted. They are reviewed by many different people with different and partial knowledge of the requirements and the representation schemes used. In the discussion which follows we examine how we can support the process of review and correcting errors in such documents.

3 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. New text processing techniques have changed many time honoured practices in this area beyond recognition. Nevertheless, as in the case of printing and graphic design, we disregard the experience that has been gained at our peril.

We may divide the processes of preparation into three broad phases, acquisition, copy-editing and proof. The acquisition phase determines if the document, as a whole, is suitable for publication. It establishes whether its content and structure are broadly suitable, if it meets the objectives set for it, if it is of an appropriate standard, and so on. If a document is deemed suitable the manuscript is then passed for copy-editing. Copy-editing ensures that the document is consistent, that structural flaws are eliminated and that an appropriate presentation is agreed. The copy editor will pay attention to ambiguities, repetitions and possible inaccuracies. Proof involves the preparation of the manuscript for printing, setting the print style, checking for typographical errors, and so on.

In this section we will largely be concerned with copy-editing though we will also draw on techniques of proof correction.

Copy-editing begins once the manuscript has been placed before the copy-editing department by the acquisitions editor. The manuscript will be read through by the copy-editor who may perform three types of editing. The first is syntactic, which covers such routine matters as spelling and hyphenation. The second is style editing, which covers both "press style", for example use of quotations, presentation of bibliographies and indexes, and the more familiar "literary style" or means of expression, for example sentence structure, choice of words and so on. The third is substantive editing, which covers rewriting and reorganising the text.

The copy-editor will mark the edits on the manuscript in a blue pencil. Different colour pencils are used to distinguish who is marking what - there are accepted conventions governing the colours used. Author corrections or queries are marked in a black pencil, printer corrections of printer errors or general queries are marked in a green pencil, author or editors corrections of printer errors are marked in a red pencil.

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. Marginal marks are essential in a large document so that the document can be rapidly scanned for edits, it is however common to omit marginal marks during the early editing stages where the editing is very heavy. There are special marks for types of error which frequently occur, such as character transposition.

Tables 1 & 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.

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* 	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

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			The vertical strokes are made through characters or words to be transposed N

Table 2

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.

this

B. Keyset Reception Device (*KRD*) - ~~the KRD~~ receives the digits that constitute the dialled number.

C. Register Functions (*RF*) - this block stores and analyzes the dialled number. The RF block recognises the expected number of digits and consults tables for selection of outgoing routes. It also supervises the set up phase of a telephone call.

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. See for example Table 3.

Instruction	Text mark	Margin mark	Note
Change to Greek letter	*circle character*	Gk	O
Use superior	*under character*	*under character*	P

Table 3

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.

All changes to the document are carried out in sequence, to avoid the problems of inconsistent changes, this is usually ensured by the simple

expedient of having a single master document. Both copy-editor and author keep copies of preceding versions. Edits can be cancelled by placing the word *stet* by the original margin mark.

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 (depending on the printing technique there may be many of these - galley proofs, page proofs, and so on). The proof will undergo some editing and revision by both author and copy-editor. The edits will, at this stage, mostly concern errors introduced during typesetting. A rule of thumb is that a maximum of 10-15% of the document may be changed after the initial proof version, any changes above this amount will probably result in considerable additional cost.

Some specialist marks to denote common printers errors such as marks from visible paper edges are used, see Table 4: Q. Generally the printer will find some problems with the manuscript and will mark these on the proofs returned to author and copy-editor using, for example, mark R.

Instruction	Text mark	Margin mark	Note
Remove extraneous marks	*circle marks*		Q
Refer to appropriate authority anything of doubtful accuracy	*circle words*		R

Table 4

Increasingly automated approaches are being adopted to marking formatting during document preparation. An example of these are generalized markup schemes in which the document structure and attributes are described and subsequently associated with processing instructions. 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.

4 Analysis

If we consider applying some of the methods discussed above to support review, annotation and correction of specifications the use of copy-editing and proof correction marks appears particularly promising.

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 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.

5 Model of Specification Construction and Review

To develop a scheme with the properties outlined above we require a model of specification construction and review to underpin it. We have developed such a model 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 & 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 requirements 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 it's 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.

6 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 can envisage a variant notation which could be used for diagrammatic specifications - Appendix A). We are not particularly attached to our notation, and it is intended as illustrative, though some attention has been given to its design. Further research attention will be paid to the human factors issues of the notation.

As a starting point 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 (akin to the speaker/hearer roles in dialogue) 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 of the scheme which follows.

For each of the locution types outlined above there is an equivalent in our scheme, see Table 5. We have employed the visual and graphic principles underlying existing mark schemes. In the manner suggested by the copy-editing marks there is a text mark with a margin mark for clarification. A question (mark i), is indicated by a line underneath the questionned statement. The filled triangle indicates it's 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.

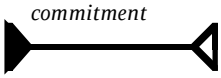
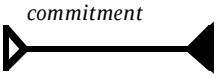
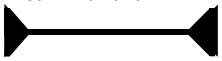
Instruction	Text mark	Margin mark	Note
Question	*under text to delimit commitment* 		i
Denial	*under text to delimit commitment* 		ii
Assert		 *followed by for example* 	Insert the statement indicated in the margin. The relevant statement should be supplied with the corresponding letter marked on it 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 5

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?".

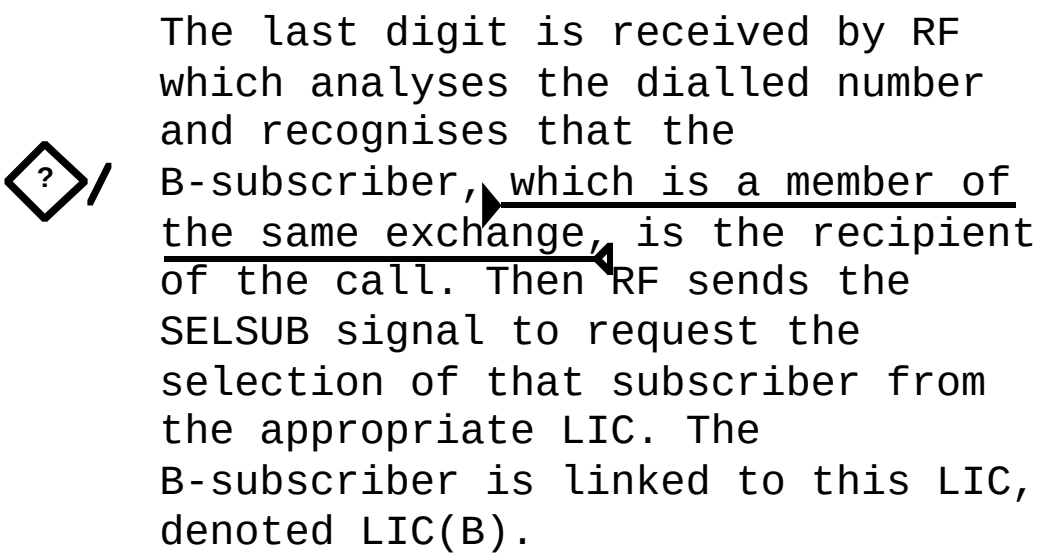


Figure 2

Figure 3 gives a small fragment of text marked up as if reviewed. The interpretation of the marks should be relatively clear. Some features may be of interest. The two resolution marks are numbered - this is to take care of the case where the commitments to be resolved are widely scattered across the text with other marks intervening. The margin marks in these cases may also be accompanied by physical text references (such as section or line number to make finding the other marks easier). The resolve consequences mark also includes a letter reference to the statement against which the marked statements are being checked.

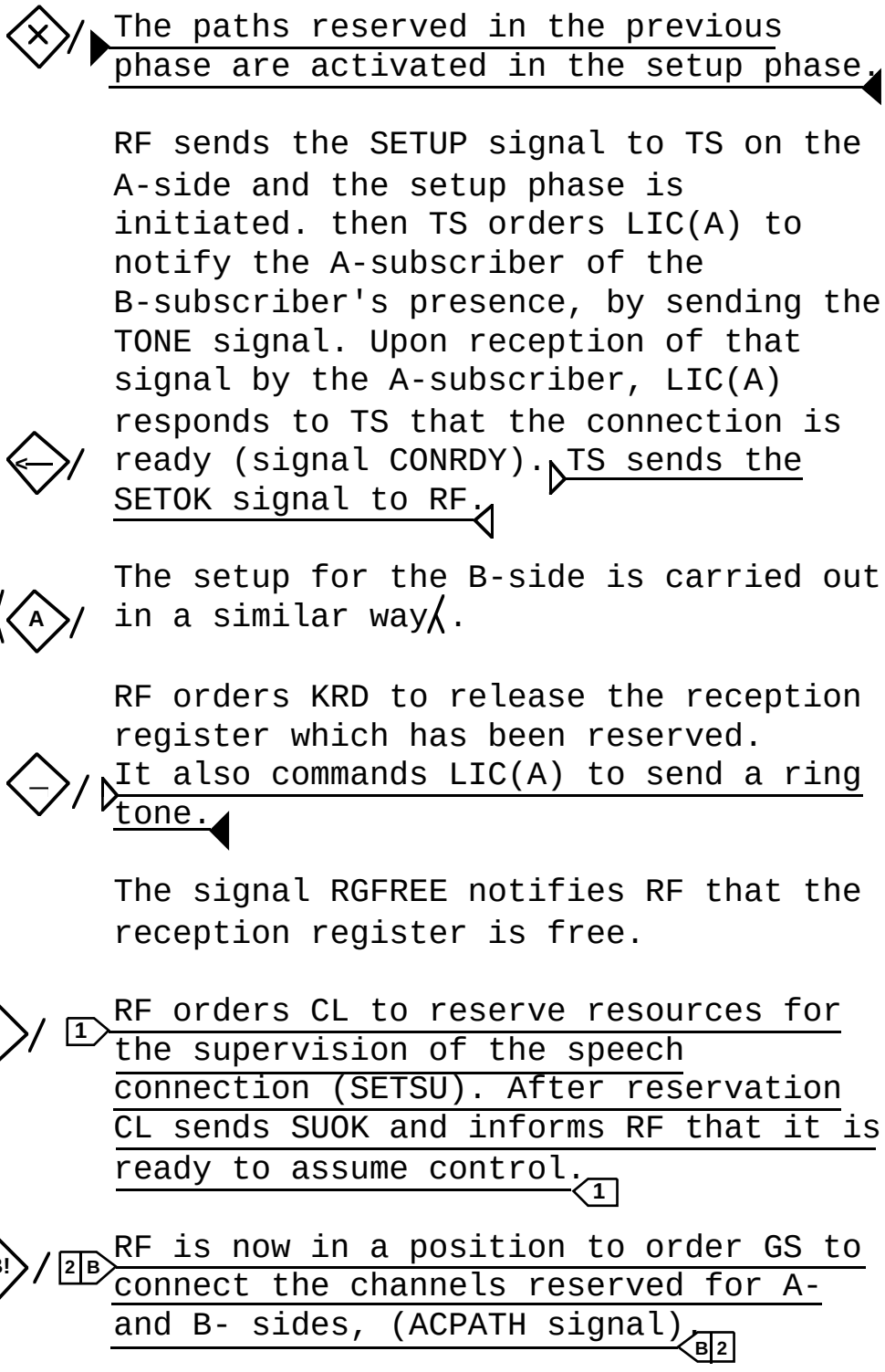

The paths reserved in the previous phase are activated in the setup phase.
RF sends the SETUP signal to TS on the A-side and the setup phase is initiated. then TS orders LIC(A) to notify the A-subscriber of the B-subscriber's presence, by sending the TONE signal. Upon reception of that signal by the A-subscriber, LIC(A) responds to TS that the connection is ready (signal CONRDY). TS sends the SETOK signal to RF.
A
The setup for the B-side is carried out in a similar way.
RF orders KRD to release the reception register which has been reserved. It also commands LIC(A) to send a ring tone.
The signal RGFREE notifies RF that the reception register is free.
!
1
RF orders CL to reserve resources for the supervision of the speech connection (SETSU). After reservation CL sends SUOK and informs RF that it is ready to assume control.
1
B!
2 B
RF is now in a position to order GS to connect the channels reserved for A- and B- sides, (ACPATH signal).
B 2
A
, the only difference being that the ring signal here substitutes for the TONE signal
B!
GS sends the PATHRDY signal and the connection between JTC(A) and JTC(B) is now established.

Figure 3

There are some simple syntactic rules governing the application of these marks. The marks for questions, denials, assertions, and withdrawals and challenges must not overlap. The marks for resolutions can overlap with questions denials and assertions. In these circumstances the statements to be checked are the ammended versions of the original statements. See for example Figure 4 in which the mark indicates that the statement to be resolved includes the assertion.

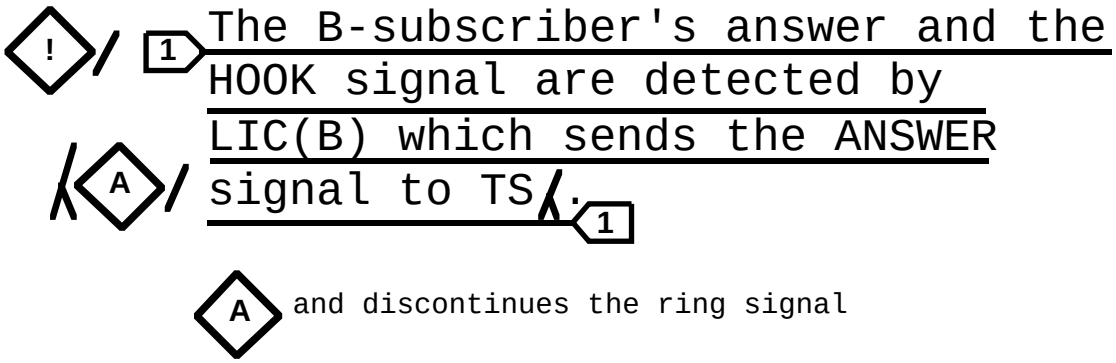


Figure 4

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, after the questioning of a statement, the appropriate response must be either the assertion (confirmation) of that statement, it's withdrawal or it's denial. Or, 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.

Assertions add a commitment on the part of the specifier to the assertion and also have the effect of committing the reviewer. The same is true for denials. Questions have no effect on the commitments of the specifier nor of the reviewer. Demands to resolve commitments also have no effect on commitments. Demands to resolve consequences have no effect on commitments of either specifier or reviewer, the given statement is treated as a fact to be assumed solely for the purpose of consistency checking. A withdrawal removes a commitment to the withdrawn statement on the part of the specifier but remains a commitment of the reviewer. Challenges also have the effect of removing the commitment of the specifier to that statement, the challenged statement becomes a commitment of the reviewer. Unless a reviewer marks a piece of text it can be taken that it has been accepted and is treated as an assertion.

There is a type of special commitment rule which we call an argument form, our scheme only includes one of these. If the response to a challenge (for example in Figure 3, a challenge to the statement "TS sends the SETOK signal to RF") is a statement given as a ground for the challenged statement (for example,"the setup for the A-side is now complete") both the specifier and the reviewer are committed to the response ("the setup for the A-side is now complete") and to the latter logically implying the former ("the setup for the A-side is now complete *hence* TS sends the SETOK signal to RF").

We have given an outline of the marking scheme and the role of dialogue and commitment rules in defining the effect of the marks and permitted responses. In the section which follows we show how the scheme can be applied.

7 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". 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.

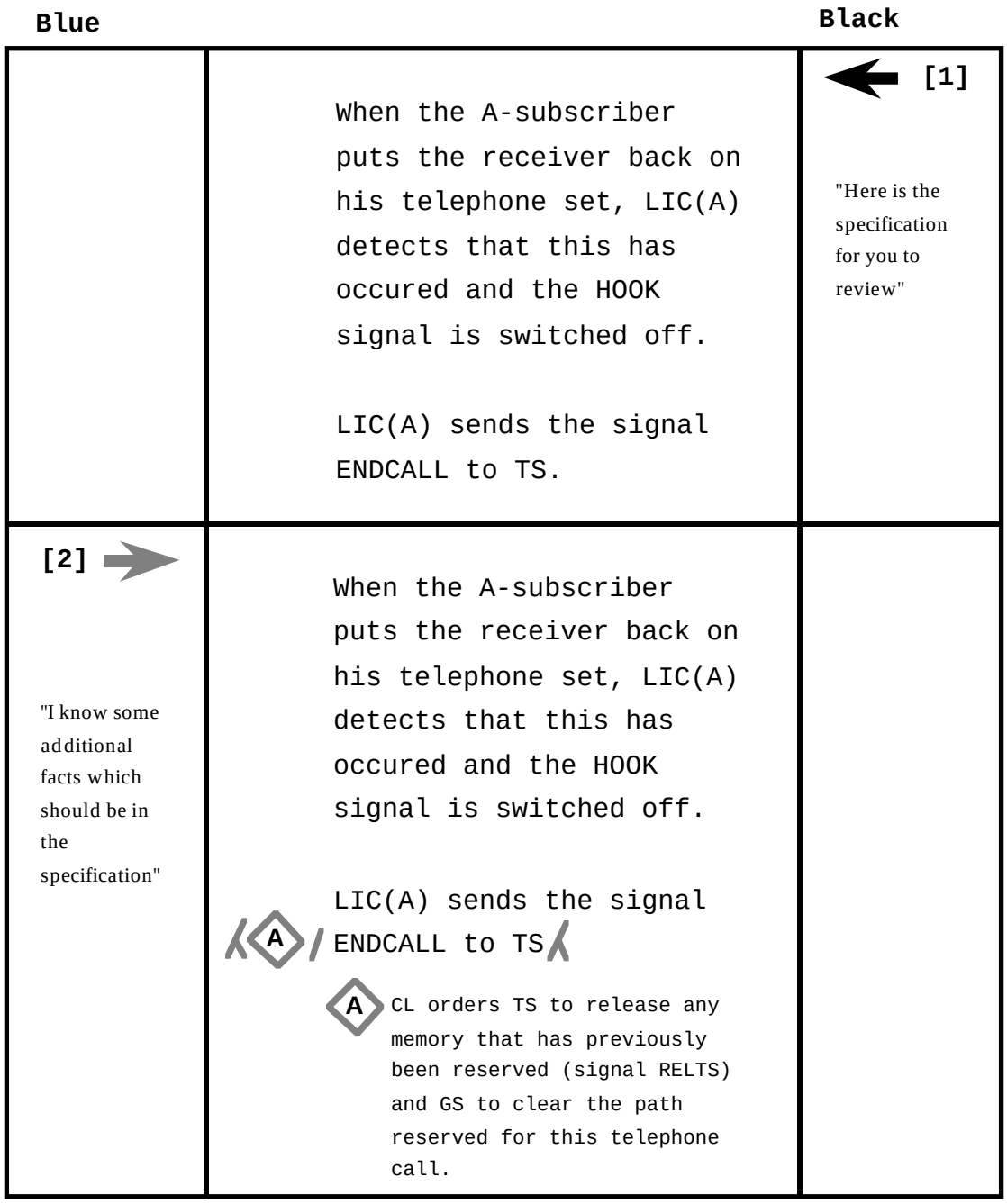
In such a manner corrections bounce back and forward until there are no further outstanding marks and all changes to the specification have been agreed. There may of course come a point where one side or another will have to exert authority to ensure that this is the case. A simple fixing sequence is shown in Figures 4 & 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 a 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 thus assertions and denials lead to direct changes while questions and demands to resolve consequences or commitments have no effect on the specification. Withdrawals and challenges are somewhat more complex. In the case of a withdrawal the commitments are reinstated (the mark is removed and no change made to the specification) unless the marked withdrawal has been immediately overlayed by another withdrawal in which both marks are removed and the text excised. Challenges are marked indelibly, the mark persists and cannot be removed from the specification. This is to avoid circular arguments.

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].


Blue
Black
When the A-subscriber puts the receiver back on his telephone set, LIC(A) detects that this has occured and the HOOK signal is switched off.
LIC(A) sends the signal ENDCALL to TS.
[1]
"Here is the specification for you to review"
[2]
"I know some additional facts which should be in the specification"
When the A-subscriber puts the receiver back on his telephone set, LIC(A) detects that this has occured and the HOOK signal is switched off.
LIC(A) sends the signal ENDCALL to TS
A
A
CL orders TS to release any memory that has previously been reserved (signal RELTS) and GS to clear the path reserved for this telephone call.
Figure 4

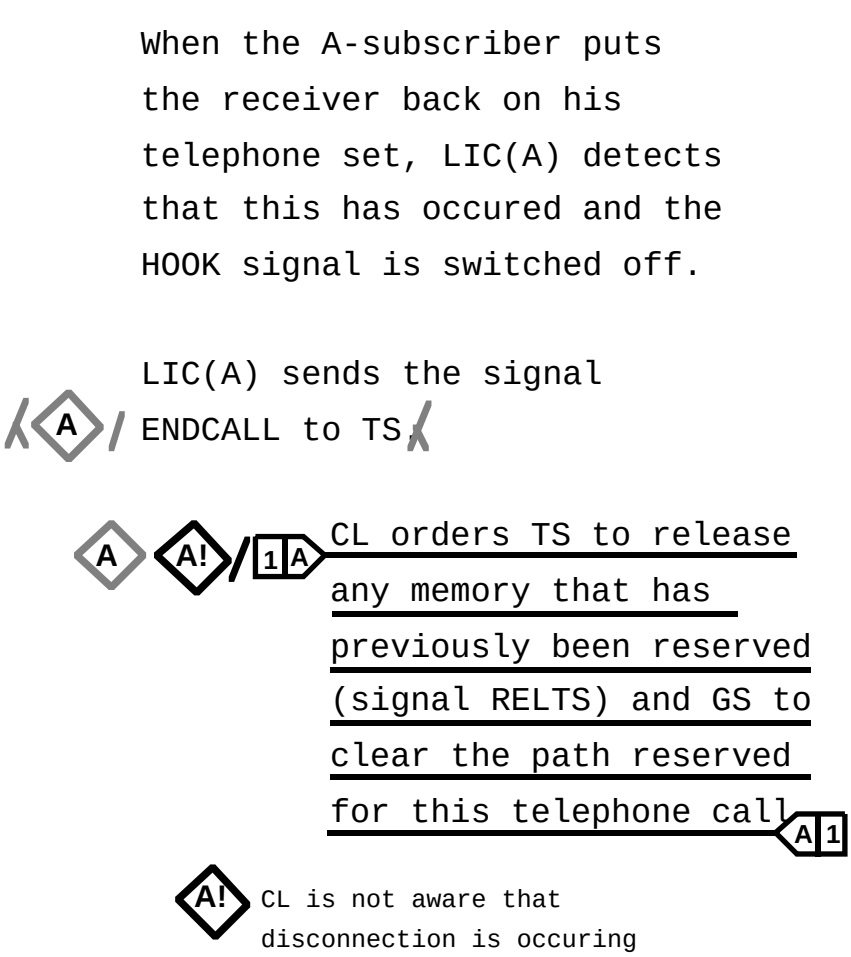

When the A-subscriber puts the receiver back on his telephone set, LIC(A) detects that this has occured and the HOOK signal is switched off.
LIC(A) sends the signal ENDCALL to TS.
A
A
A!
1
A
CL orders TS to release any memory that has previously been reserved (signal RELTS) and GS to clear the path reserved for this telephone call
A
1
A!
CL is not aware that disconnection is occuring

Figure 5

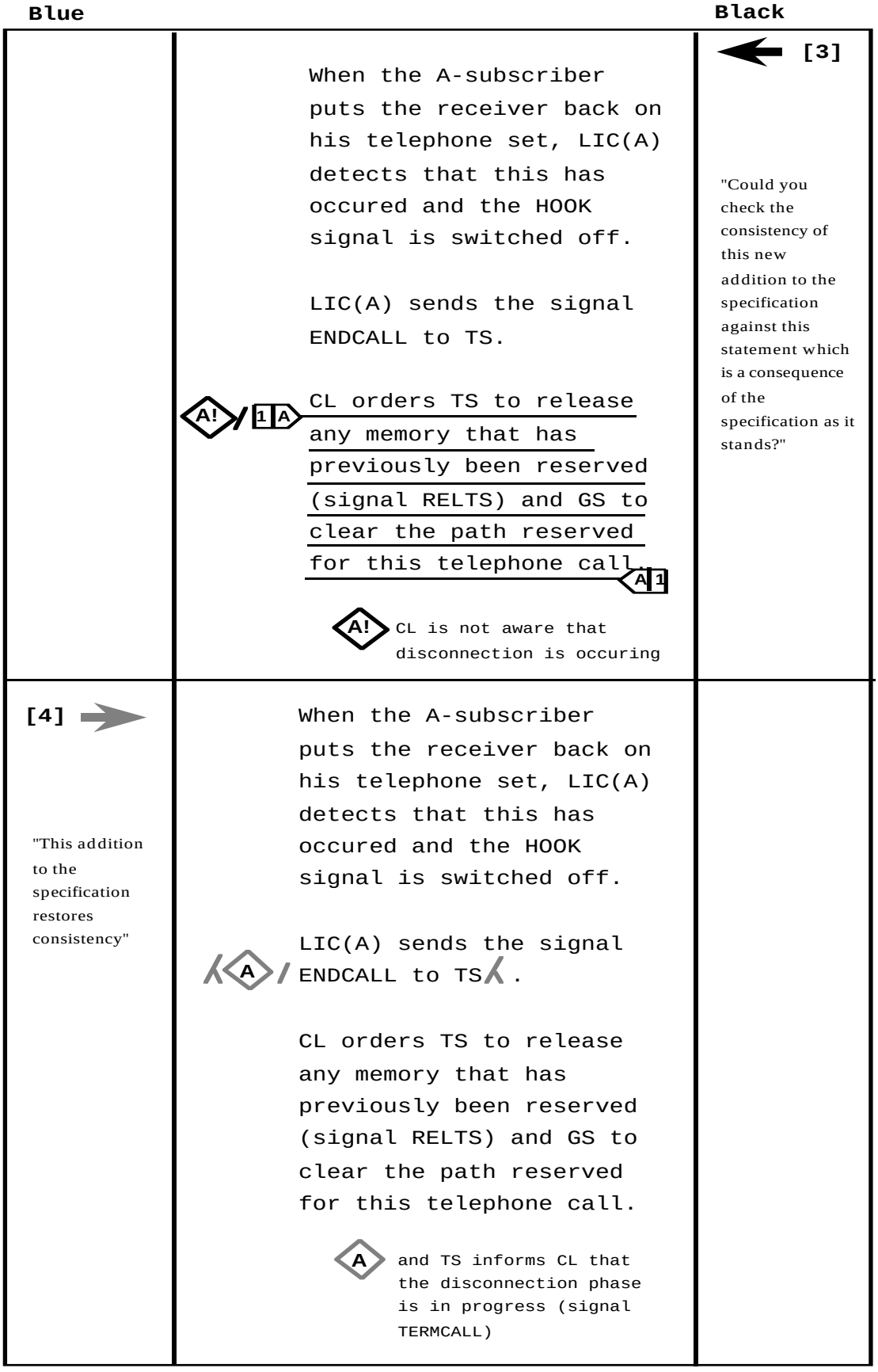

Blue
Black
[3]
When the A-subscriber puts the receiver back on his telephone set, LIC(A) detects that this has occured and the HOOK signal is switched off.
LIC(A) sends the signal ENDCALL to TS.
A! 1 A
CL orders TS to release any memory that has previously been reserved (signal RELTS) and GS to clear the path reserved for this telephone call.
A 1
"Could you check the consistency of this new addition to the specification against this statement which is a consequence of the specification as it stands?"
A!
CL is not aware that disconnection is occuring
[4]
"This addition to the specification restores consistency"
When the A-subscriber puts the receiver back on his telephone set, LIC(A) detects that this has occured and the HOOK signal is switched off.
A
LIC(A) sends the signal ENDCALL to TS.
CL orders TS to release any memory that has previously been reserved (signal RELTS) and GS to clear the path reserved for this telephone call.
A
and TS informs CL that the disconnection phase is in progress (signal TERMCALL)

Figure 6

8 Experience and Tool Support

We believe that our scheme is of value to discipline and make systematic review, annotation and corrrection. We have gained experience on a significant number of small examples and are currently working on larger "real-life" 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.

9 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 & Salaun (1987) also Welbourn & 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 & 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 & 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).

10 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. Further, by basing our review scheme on a formal model of negotiation and specification construction we can preserve a rationale for the specification and maintain traceablity of requirement during reviews.

Acknowledgements

Thanks to my colleagues and students, in particular thanks to Hugo Fuks, Jeff Kramer and Jeff Magee. Thanks also to the CEC and Esprit project REX (2080) who provided the funding for this work. The examples are extracted from Votsis & Papoulias (1989) and Reed (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, pp533-539; IEEE CS Press.

Feather, M. (1987); Constructing Specifications by Combining Parallel Elaborations; [To appear] IEEE Trans. Software Engineering.

Fickas, S. & Nagarajan, P. (1988); Being Suspicious: critiquing problem specifications; Proc AAAI 88; 1, pp19-24; Morgan Kaufmann Pub.

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

Gould, J. & 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. & Whitehead, K. (1987);"Tool Assisted Requirements Analysis: TARA final report"; Imperial College, Dept. of Computing, Technical Report 87/18.

Reed, J. (1989); A Design Specification of a Telephone Switch Using CSP & Z; Private Communication.

Reiss, S. (1990); On the Use of Annotations for Integrating the Source in a Program Development Environment; In: Finkelstein, A. Tauber, M. & Traunmuller R. (Eds) Human Factors in Information Systems Analysis and Design; North Holland, pp25-36.

Sharples, M. & Pemberton, L. (1990); Starting from the Writer: guidelines for the design of user-centred document processors; University of Sussex, School of Cognitive & Computing Sciences Technical Report, CSRP 154.

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

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

Welbourn, L. & Whitrow, R. (1988); A Gesture Based Text Editor; HCI'88 - People & Computers IV, Jones & Windsor (Eds); (full reference not available).

Appendix A

The notation we have shown apply primarily to natural language and formal representation schemes. An alternate (but semantically equivalent) set of marks can be devised for graphical schemes. An off-the-cuff example of the use of such marks is given below. Note that we can mark a resolve consequences between a graphic element and a piece of formal notation.