The Knowledge Engineering Review, Vol. 12:3, 1997, 249–270

Knowledge engineering in the communication of information for safety critical systems

CORIN A. GURR

Human Communication Research Centre, University of Edinburgh, Edinburgh EH8 9LW, UK, email: C.Gurr@ed.ac.uk

Abstract

The design and assessment of safety critical systems often involves broad and distributed teams of designers, suppliers and analysts who represent diverse areas of expertise and motivations. Accurate and effective communication between these groups is therefore an issue of primary importance. The formalisation of specifications and arguments of safety can be of significant benefit in ensuring the consistency of evidence in such cases, when it must be presented across many domains. However, a formal description of a safety critical system may be unconvincing unless it is presented in a form which is (or forms which are) accessible to the broad range of users and assessors of safety cases. This raises issues of human communication which include the tailoring of information to particular communicative tasks; the efficacy of differing media for communication and the cognitive impact that such differing media have. This paper draws together work in fields of knowledge engineering, knowledge based systems and human communication in an effort to address, from a sound theoretical basis, these and other communication issues raised by the use of formal descriptions in safety critical systems. Further, this paper argues that a primary role for knowledge based systems techniques in safety critical systems is in supporting the communication of information.

1 Introduction

A significant difference between safety critical systems and non-safety critical systems is in the degree of attention paid to assessment and testing, often enforced by industrial standards, both of the design and the design process of such systems, and of the system in use. Software can be considered to be safety critical only in the context of the larger system: of the hardware in which it is implemented, the attached peripherals and the operating environment. An error in software cannot directly cause injury, but the consequences of a software error upon the attached peripherals or operating environment—that it causes a missile to be launched unintentionally, prevents the brakes of a aircraft from operating when it lands, or fails to correctly shut down a nuclear reactor in an emergency situation—may cause injury, fatality or major environmental damage. When considering the safety, correctness and dependability of safety critical software we must therefore consider the software in the context of the whole system and the environment in which it operates. This entails drawing upon a body of expertise from domains as diverse as electrical, chemical, automotive, aerospatial and nuclear engineering; risk assessment; and environmental analysis. Consequently, the design and assessment of safety critical systems often involves broad and distributed teams of designers, suppliers, users and analysts who represent diverse areas of expertise and motivations and a broad range of technical and non-technical disciplines. Accurate and effective communication between these groups is therefore an issue of primary importance.

To be of interest to an audience of the size and diversity common in the design and assessment of safety critical systems, any description of the system or its design process would inevitably contain a great deal of information. For a given reader much of this information might be irrelevant, although all of the information would likely be of relevance to some reader. Naturally therefore there is a

significant advantage in being able to give multiple presentations of this information, each of which presents some part or aspect of the information available. From these a particular reader would be able to select only those presentations of interest to them.

We commonly see multiple representations being used in the specification of a particular system, with differing representations being chosen to emphasise differing *aspects* of the system under consideration. For example, hardware designers mainly use three graphical representations when describing a piece of hardware: circuit diagrams, state diagrams and timing diagrams. In this way several different views may be provided for a given system. Each view will be chosen for its merit in displaying a particular aspect of the system under consideration in a manner which emphasises all pertinent details while abstracting away from all irrelevant details. A key issue in any specification which uses multiple representations is that the information expressed should be consistent across all representations. One means of achieving this is by using formal descriptions of the system or design process as the basis from which multiple representations may be presented.

The use of formal techniques can be and have been, it has been argued (Bowen and Stavridou, 1993a, b), highly advantageous in both the design and assessment of safety critical systems, both for modelling systems and for representing arguments of safety. Henceforth we shall use the term "formal description" to refer to the joint notions of formal specifications and formal arguments. Formal representations assist by removing ambiguities in specifications and arguments, and by making explicit assumptions which might otherwise remain hidden. However, a formal description of a safety critical system may be unconvincing unless it is presented in a form which is (or forms which are) accessible to the broad range of users and assessors of safety critical systems.

Attempting to achieve an acceptable presentation of a formal description raises issues of human communication to address the visualisation of formal languages and arguments. These issues include:

- the tailoring of information to particular communicative tasks;
- the efficacy of differing media (for example, textual and graphical) for communicating information;

For the latter point we note that assessing the appropriateness and efficacy of a given media for communicating information cannot be done without regard for the abilities and preferences of the human reader to whom we are communicating. Consideration must be given not only to the computational expressiveness of a given media or notation, but also to the cognitive impact that different media have upon human reasoning.

The field of knowledge engineering provides a substantial body of research on the first of these issues and the experience gained from constructing interfaces to Knowledge Based Systems (KBS) can lend much insight into the second. Further insight into this second point may be gleaned from research into cognitive aspects of human communication. Such research also supplies a sound theoretical basis for the latter issue: an understanding of the impact of differing media upon human understanding and reasoning.

This paper draws together work in the above fields of knowledge engineering, knowledge based systems and human communication in an effort to address, from a sound theoretical basis, the communication issues raised by the use of formal descriptions in safety critical systems. This paper has four main parts. Section 2 presents a discussion of why we need to make formal descriptions accessible and of why it is that the most important issue here is of communicating the assumptions which underly formal descriptions. Section 3 discusses the needs for, and advantages of multiple presentations of formal descriptions, utilising multiple media, for achieving this communication of underlying assumptions. In section 4 the implications of industrial standards upon the use of KBS in safety critical applications is addressed. At first glance it would appear that these standards exclude the use of KBS, yet on closer examination it may be seen that an important and appropriate role for KBS techniques lies in the structuring and presentation both of arguments of system safety (known as "safety cases"), and of the industrial standards themselves. A review of such KBS techniques and theories of human communication, indicating how these may be used in the

organisation and generation of multiple presentations, is given in Section 5. Finally, the paper ends with some conclusions and suggestions for further reading.

2 Formal descriptions: the need for validation

To make both a formal description and the consequences of it convincing they must be validated to the satisfaction of the reader. For a given conclusion or argument the reasoning steps which led to it must be accessible to the reader (that is, presented in a format they are either familiar with or which is clearly and easily understood). Furthermore, given that any formal description is necessarily an abstraction based upon certain assumptions, the accuracy of the model underlying this description must also be validated to the satisfaction of the reader. Thus the assumptions which permit the abstractions of the model must also be clearly identified and made accessible to the reader. The hypothesis taken here is that it is this identification and communication of the assumptions underlying a formal model which is the most important communicative task. Communicating such assumptions firstly permits their validation, and thus the validation of the formal description itself, and secondly provides a sense of *context* which assists a reader in interpreting the consequences of a given formal description.

The effectiveness of suitable presentations of formal descriptions is particularly well illustrated by the analysis of a case study into the use of formal methods carried out by industry. A formal model of a safety-critical multiprocessing system was derived from its natural language specification. The designer's examination of their formal model uncovered a potential behaviour of the system which they had considered impossible. A further analysis by the specifiers showed that they had made certain assumptions concerning the relative timings of concurrent agents within the system and that these assumptions were not explicitly recorded, were not necessarily justifiable and had not been identified by peer review. Subsequent analysis of this case study, described in detail next, illustrates firstly the effectiveness of different presentations in communicating assumptions underlying the formal model. Secondly it demonstrates the importance of presenting these assumptions, as one can be shown to be incorrect (thus invalidating one aspect of the formal model) and a second is crucial in explaining a significant consequence of the model (regarding system behaviour) which was considered unusual by its designers.

The system in question, referred to as a "Twin Buffer" system, consisted of two buffers which could independently receive data and a third "Voter" component which responded to "vote" requests by reading the contents of each buffer and outputting the results of a computation of these. The industrial designers constructed a formal model of such a system in the language CCS (Calculus of Communicating Systems) (Milnes, 1989), an algebraic formal language for modelling concurrent systems. Subsequently an independent study (Gurr, 1995) analysed the CCS model in question, constructing and assessed a number of different representations of it, including two diagrammatic representations. Several errors were uncovered in this process. The CCS model had already proved its effectiveness to its designers by showing that an important safety property had not been captured, although the reason for the lack of this property in the model was initially far from clear. By contrast, one diagram clearly highlighted the mistaken assumption by the original designers which was the root cause of the failure of the model to verify the required property. Furthermore, two unrecorded and mistaken assumptions were immediately obvious when the diagrammatic representations of the formal model were assessed.

The first diagrams in question are the "flow graphs" shown in Figure 1, which make immediately apparent an unusual feature of *Twin*, the CCS model of the twin buffer system, that it never produces any requested output. A flow graph represents the potential lines of communication, represented by labelled arcs, between components of a CCS specification, represented by named circles. The flow graph of Figure 1(a) hides the interior details of the specification, showing only the potential lines of communication from the twin buffer system to the outside environment. The flow graph of Figure 1(b) represents the same specification, but shows also the interior components, the two buffers and the voter, and the potential internal communications between them. The diagram of

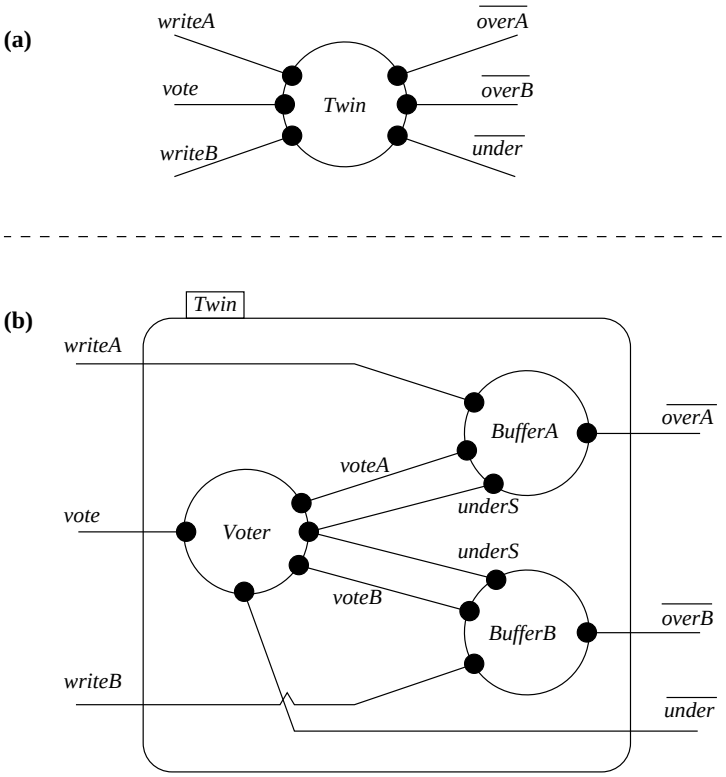

Figure 1 "Opaque" (a) and "Exploded" (b) views of flow graph for CCS specification

Figure 1(a) is particularly pertinent as it makes immediately obvious the fact that the only outputs potentially produced by this model of the Twin system are the warning messages $\overline{overA}$, $\overline{overB}$ and $\overline{under}$ (warnings of buffer overflow and underflow). What is unusual is that despite input being written to the buffers (by *writeA* and *writeB* actions) and requests for data (*vote* actions), the actual results of any request for data are not represented.

While the above feature of "no data" is certainly an oddity, and illustrates particularly well the efficacy of representing limited aspects of information by diagrams (static structure only, in this case), we note that this error has arisen from an over-abstraction in the construction of the CCS specification, rather than being an error inherent in the twin buffer system. A deliberate decision was made to abstract from a representation of any actual data in the twin-buffer system while maintaining a representation of the situation where the system failed to manipulate the data. Unfortunately the specifiers abstracted too far by failing to capture any explicit representation of the system succeeding in manipulating the data. The absence of any modelling of data does, however, have an affect on the modelling of the *Voter* component which, as we show next, can be identified as an error.

Figure 2 presents the CCS specification of the voter component of the twin buffer system. This specification indicates that the voter responds to a *vote* action by outputting a $\overline{voteA}$ and $\overline{voteB}$ action and then moving into the *Votelock* state. *Votelock* was designed to "provide a mechanism for deadlocking the *Voter* process when either of the buffers deadlocked" (which would be indicated by an *underS* warning). In CCS the summation connective "+" represents non-deterministic choice,

$$Voter \quad \stackrel{def}{=} vote.\overline{voteA}.\overline{voteB}.Votelock$$

$$Votelock \quad \stackrel{def}{=} underS.\overline{under}.Trip + Voter$$

Figure 2 CCS specification of the voter component

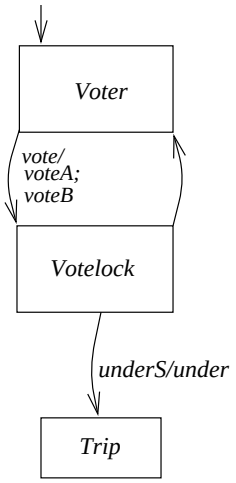

Figure 3 Statechart of the CCS specification of the voter component

indicating that *Votelock* can either respond to an *underS* action by outputting an $\overline{under}$ warning and then moving into the *Trip* state (deadlocking *Voter* as required), or it can return to the *Voter* state and await further *vote* actions.

The above CCS definition of the voter component is indicative of certain assumptions, which may be described as "design guidelines" informally adopted by the specifiers. These assumptions were captured graphically in the subsequent study (Gurr, 1995), where the CCS specification was translated into a "statechart". Statecharts (Harel et al., 1986) are a graphical language for representing reactive (concurrent) systems, and Figure 3 is the statechart derived from the CCS agent *Voter*. In statecharts states are represented as labelled blocks and transitions as arrowed arcs with non-empty labels of the form a/t, which denotes a transition triggered by the event a that generates the sequence of actions t. The unanchored arrow pointing to the *Voter* state is used to indicate that this is the initial state for this component.

The most significant of the design guidelines (assumptions) informally adopted by the specifiers is that certain states were deemed somehow significant and that sequences of actions which connected these states should always be of the form *action*/*reaction*. An example of this is the $vote/\overline{voteA}.\overline{voteB}$ sequence which connects the *Voter* and *Votelock* states. We have capture this assumption precisely in the graphical language of statecharts, as indicated by Figure 3. In that statechart only the significant states (those named by the specifiers) are explicitly represented and transitions along arcs are of the form *action*/*reaction*, in accordance with both the syntax of statecharts and the specifiers assumption.

A peculiarity of this statechart is that one of the two arcs between the *Voter* and *Votelock* agents has no label. This is in fact an error in this specification as it indicates that the *Votelock* agent is able to move into the *Voter* state even if an *underS* warning is received. If this should happen then it is possible for this model of the twin buffer system to subsequently reach an erroneous deadlocked state. This error is connected with the previously mentioned error of no output data being represented in the CCS specification, as the specifiers intention had been that the unlabelled arc should represent the transition in states to be made if data was successfully read from both buffers. Had this data been explicitly represented then this would have given a label to the arc in question and thus given the CCS specification the correct behaviour (successful retrieval of data and the warning message being mutually exclusive).

The specifiers of the full natural language specification of the twin-buffer system specification recorded eight potentially safety-critical properties which it was desirable to demonstrate held of the system. The last of these properties was that "failure of either input process or the voting process will result in the correct trip position" (and the system reporting either an overflow or underflow

error, as appropriate). However, animation of the CCS model by an automated verification tool showed that in certain circumstances a sequence of actions which the designers had expected to cause a trip (and the reporting of an error) could be performed adequately by the system *without* causing either a trip or an error to be reported. After examining both the formal specification and the natural language specification in more detail, both specifiers agreed that the sequence of actions in question was correct with respect to both specifications and that it was the specifiers assumption that it would cause a trip which was incorrect.

A significant advantage of using the statechart of Figure 3 to represent the CCS specification is that it provides us with a readily understandable formal account for the underlying, undocumented, assumption which led the designers to their erroneous expectations of system behaviour. This assumption, while not expressed explicitly, was implicit in the specifiers intended interpretation of the CCS specification. The specifiers undocumented assumption can be explained as being a belief that an action, such as *vote*, and all of the actions associated with it (that is, $\overline{voteA}$ and $\overline{voteB}$) would be processed sufficiently quickly that no intervening actions could occur. The syntax of the CCS specification indicates which are associated sets of actions in the specifiers intended interpretation. Examining the statechart of Figure 3 we see that this assumption corresponds precisely to the assumption that transition along an arc, such as that labelled $vote/\overline{voteA}.\overline{voteB}$, will happen sufficiently quickly that no intervening transitions can occur. This assumption is formally expressed in the semantics of statecharts and is known as the *Synchrony Hypothesis* (Pnueli and Shalev, 1992). However, it is an assumption which is not matched by the semantics of CCS, in which it is possible for intervening actions to be processed in between the processing of actions such as *vote* and $\overline{voteA}$. This capability meant that the CCS specification did not always behave as the specifiers had expected. On review it was decided that the behaviour of the CCS specification was correct, and that it was the specifiers initial assumption which was at fault.

We may manipulate this formalisation of the specifiers undocumented assumption to describe the unexpected behaviour they observed. By suggesting that transitions may intervene between the various actions in a single transition (for example, between the three actions in the transition $vote/\overline{voteA}.\overline{voteB}$) we are able to clearly indicate how and where the specifiers undocumented assumption can be violated. This illustrates how a suitable representation, the statechart in this case, can be used as a mechanism for capturing an underlying assumption in some formal description and holding it up for scrutiny.

3 Multiple presentations: making formal descriptions accessible

Formal descriptions of safety critical systems need to be accessible to users and assessors from a broad range of technical and non-technical disciplines and who represent a wide range of differing interests. To be of interest to an audience of this size and diversity any formal description would inevitably contain a great deal of information, including numerous underlying assumptions made by its producers. For a given reader much of this information might be irrelevant, although for any part of the information there would likely be some reader who would find it of relevance. Naturally therefore there is a significant advantage in being able to give multiple presentations of this information, each of which presents some part or aspect of the information available. From these a particular reader would be able to select only those presentations of interest to them. The advantage of this multiplicity of presentations being derived from a single formal description is the ease of guaranteeing consistency between differing presentations.

An illustrative example of the benefit of having multiple presentations of information for safety critical systems is the HAZOPS (HAZard and OPerability Studies) (HAZOPS, 1992) hazard analysis technique. HAZOPS is a technique that originated in the chemical industries which involves engineers and experts from a broad range of disciplines holding a series of "structured brainstorming" sessions to identify and assess the potential hazards of a proposed design. A typical HAZOPS is oriented around a diagram of the proposed design (for chemical plants piping and instrumentation diagrams are used). The HAZOPS team examine in turn each component depicted

on the diagram and consider the hazards and likelihood of failures or deviations from its intended function. Typically, each team member will have access to that information on the proposed design which is relevant to their field of expertise. Thus, the team is able to bring a great breadth of experience and data to the analysis yet, by coordinating the analysis around a common focus (the diagram), individual team members need not be concerned with information beyond their own area of expertise. Furthermore, the diagram used in a HAZOPS typically represents the proposed design at a general enough level to be clearly understood by all team members, regardless of technical discipline and expertise, while still being sufficiently detailed to make an analysis based upon it worthwhile. For example, any two components could safely be assumed to be independent if they are separated on the diagram, and to need analysis if they are connected. Recently there has been substantial interest in the application of HAZOPS to software, where a suitable diagrammatic representation of the code is used to focus the attention of a team of analysts. An example of such a diagram might be the flow diagram of 1(b), which represents the major components in a twin-buffer system and the lines of communication linking them which indicate data flow.

When multiple presentations of information are available the advantage of having different presentations in different media available is twofold. Firstly, the task of distinguishing between differing presentations is made significantly easier when the languages used to express the presentations differs. These languages need not necessarily be in differing media. For example, a contrast may be achieved by using two textual languages which exhibit a noticeably differing style and lexicon. The second advantage, which argues more strongly for multimedia presentations, is that differing media have differing inherent properties which make them more or less appropriate for the presentation of different forms of information. For example, graphical languages such as tables, graphs and maps are well suited for presenting certain types of information while other types of information are best presented in textual languages such as logics, equations or natural language.

One aspect of the appropriateness of differing media for communicating certain kinds of information is that of cognitive impact. Indeed, it is not only important to consider the general impact of differing media, but also that individual human cognitive differences can play a significant role in the interpretation of information. For example, a recent study of student users of Hyperproof (Barwise and Etchemendy, 1994), a multimedia (textual and graphical) logic teaching software tool, has shown that the particular cognitive abilities and preferences for particular media of students have a strong effect on their usage of automated tools.

Stenning, Oberlander and Cox (1995) report a study of Stanford undergraduates learning first order logic from Hyperproof, as compared with a control class learning from the natural-deduction-sentential-only part of the Hyperproof program. Although Hyperproof graphics are rather inexpressive relative to the predicate calculus, some ''abstraction tricks'' have to be used in order to enable the program to display a manageable number of graphical representations for classes of possible models. Figure 4 shows the Hyperproof interface. The cylinders visible on the chequer-board stand not for cylinders but for polyhedra from the Hyperproof domain. Their questions marks can be replaced by badges indicating shape (but still not indicating size). Alternatively, ''paper bag'' icons can mark the position of polyhedra, revealing their size but not their shape.

The study predicted that students' facility with using these graphical conventions in Hyperproof would determine the usefulness of the Hyperproof environment in teaching. It was expected that these devices would be well used and that the environment would be rather successful. As the cited paper shows, the prediction that it was the limited graphical abstraction conventions on which usability turned was correct. However, it was surprising just how strong were the interactions between students' prior styles of reasoning and the outcomes of different methods of teaching. For those students who were adept at analytical reasoning (based on prior tests, Hyperproof was an extremely effective way of teaching logic. For the, in general, equally able students who scored lower on the pretests, Hyperproof teaching was a disaster on some transfer tests.

This same study also indicated that, more seriously, a student's ability can actually decrease if they are limited to using a strategy or media, imposed by the tool's interface, with which they are less comfortable (Cox et al., 1994). This and similar studies indicate the advantage of offering readers a

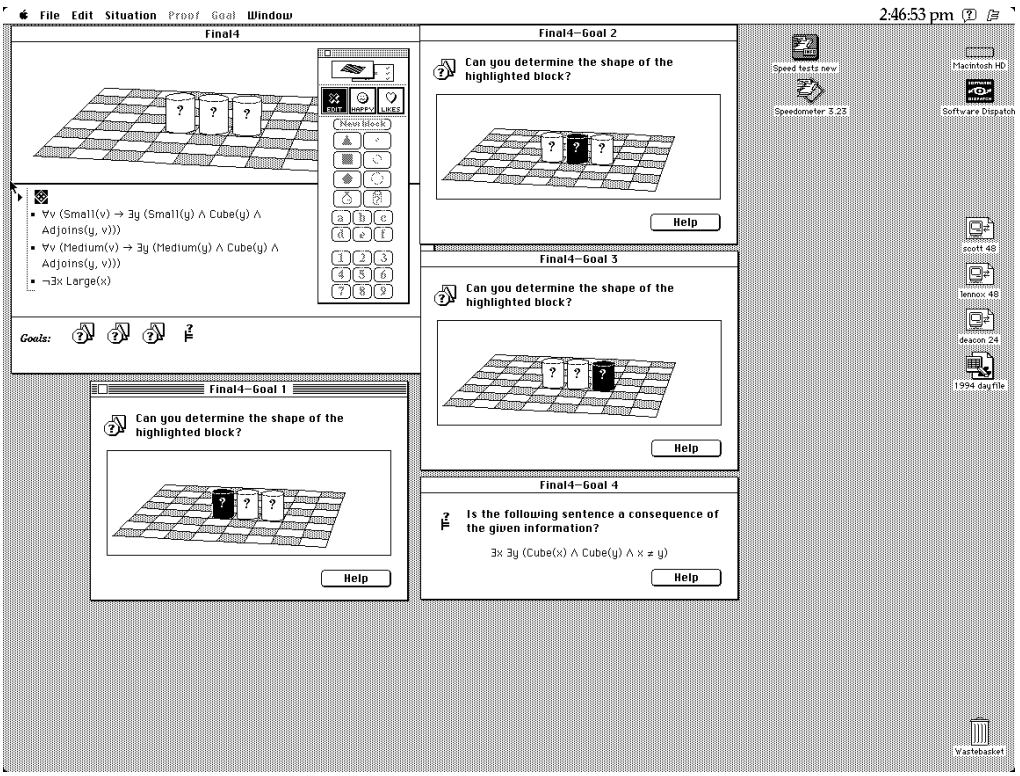

Figure 4 The Hyperproof Interface. The main window (top left) is divided into an upper graphical pane, and a lower calculus pane. The tool palette is floating on top of the main window, and the other windows reveal a set of goals which have been posed.

range of information presentation medium which they may select from on the basis of their individual cognitive preferences.

3.1 Natural language presentations

Having decided upon multimedia presentations of specifications and arguments the problem now becomes the selection of appropriate languages, or an appropriate range of languages, best suited to represent the information at hand. Hall's response (Hall, 1990) to the sixth myth of formal methods, that "formal methods are unacceptable to users", is that a formal specification must be made comprehensible to the user. Hall suggests that this may be achieved in three ways, the first of which is always essential:

- Paraphrase the specification in natural language.
- Demonstrate the consequences of the specification.
- Animate the specification.

This argument suggests that a natural language presentation is essential to make a formal specification comprehensible. Examining this argument carefully we may see that a weaker statement is possible: that a presentation in a language familiar to the user is essential. For example, in the field of electrical engineering a specification may be presented to a user via the circuit, logic and timing diagrams with which they are familiar. Thus, for specific application areas, a presentation of the formal specification may be made in any language or combination of languages (textual or graphical) which is common to that application area and for which there is a well understood semantics.

In most safety-critical applications, a formal specification must be read by users from a broad range of both technical and non-technical disciplines. Therefore, to make the formal specification accessible it is highly desirable that, in addition to presentations in a variety of special purpose notations, certain parts of it should also be presented in a suitably generic language. In the absence of a universal graphical language of any great degree of sophistication and expressiveness this implies that a generic textual language, and thus—as stated by Hall—a natural language, is necessary. The great advantage of using a natural language is that it is effectively accessible to all readers. The disadvantage of natural language is that without careful usage it can very easily be ambiguous or, at the very least, less precise than may be necessary. Theories of natural language generation and processing can help to compensate for this potential for ambiguity. Such theories are reviewed in section 5.

3.2 Graphical presentation languages

Theories of human communication developed initially for linguistic purposes, either for written text or spoken dialogues, may be applied to good effect in the development of theories for communication involving graphics. Oberlander (1996), for example, draws together the findings from several influential studies in graphical communication to argue for parallels between "pragmatic" phenomena which occur in natural language, and for which there are established theories, and phenomena occuring in graphical representations. Notable among these is the work of Marks and Reiter (1990) who relate the design and viewing of diagrams to a discourse between designer and user. In this context they invoke Grice's *cooperative principles* of discourse, four maxims which can be thought of as guiding usual communication: Quality, Quantity, Relation and Manner (these maxims are spelt out in full in Grice (1975)).

Marks and Reiter explore the possibility of applying the Gricean theory of implicature (that failing to obey the maxims can lead to unwanted implications for the reader) to suggest a reasoned approach to avoiding ambiguities or misleading information in graphical representations. For example, Figures 5, 6 and 7 are presented by Marks and Reiter and display, respectively, a network diagram of a computer's disk subsystem (with no unwanted implicatures) and two further diagrams, each of which carries unwanted implicatures which may be explained by failures to obey the Gricean

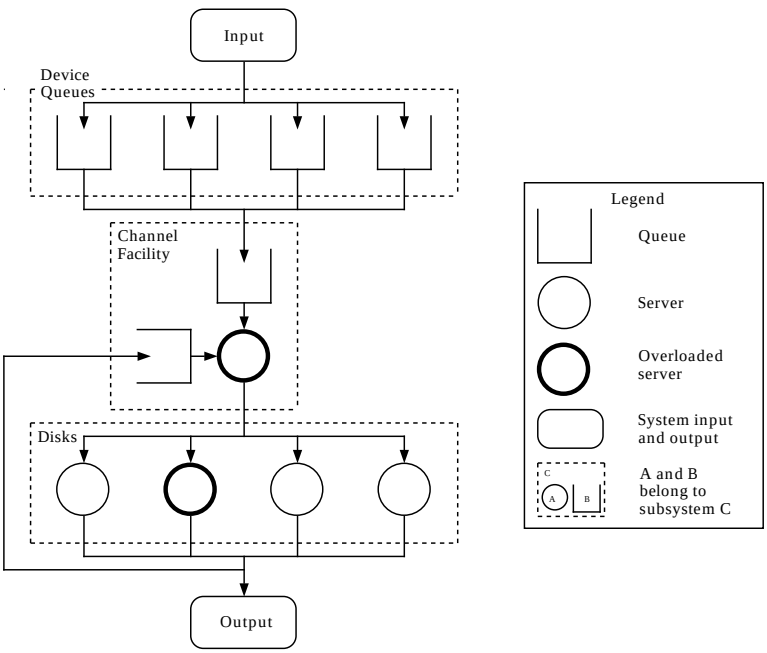

Figure 5 The disk subsystem of a computer

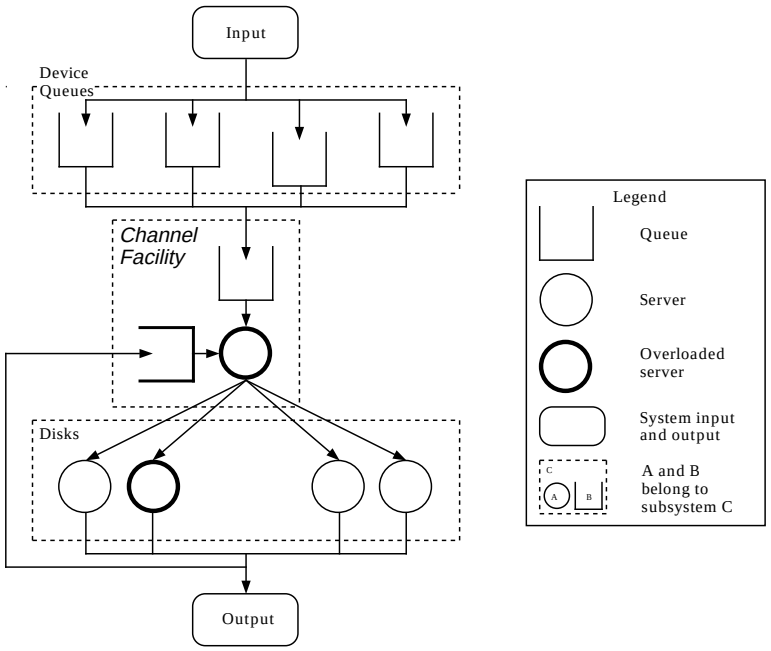

Figure 6 A variant of Figure 5, possessing unwanted implicatures

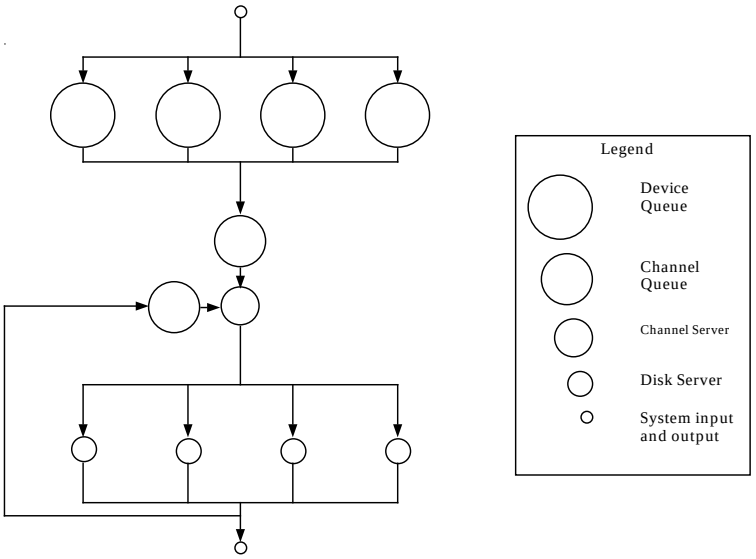

Figure 7 Another variant of Figure 5, possessing different unwanted implicatures

maxims in their construction. Figure 6 for example is misleading for at least four reasons. First, the pen-width used for the channel-queue differs from pen-width for all other queues. The unwanted implication is that this queue must be different to the others somehow, as otherwise the same pen-width would have been used. Secondly, disk-symbols are grouped as two perceptual gestalts. Again the unwanted implication is that this reflects some grouping in the domain, as otherwise they would have been placed regularly. The final two differences which lead to unwanted implicatures are that the device-queue symbols have been laid out irregularly, implying that one of them differs from the others somehow, and that a different font has been used for the channel-facility's text label, implying that it must have different sub-system status.

Note that the diagram of Figure 6, just as that of Figure 5, faithfully portrays all the features of the assumed network model. The problematic implications of the second diagram are not in fact

caused by it being incorrect, but rather by various aspects of it (unspecified in the semantics for the diagram) which are misleading. All of the above problems in the second diagram arise because there is a general expectation of regularity, and any departures from this are assumed to be meaningful. The problem with the diagram is that it contains irregularities where the domain has none and, in the case of the grouping of disk symbols, an extra regularity not in the domain. The second variant diagram of Figure 7 provides a further example of unwanted implications due to extra unwarranted regularities. In this third diagram node symbols are ordered by size, implying (incorrectly) that there must be a similar ordering relation among the vertices in the network model.

There are two lessons to be learned from the above diagrams. First the diagram of Figure 7 indicates the importance, in designing a graphical language, of selecting appropriate notations (without inherent and misleading implications) for the information to be represented. Secondly, as indicated by the diagram of Figure 6, the importance of considering the potential perceptual impact of secondary aspects of a diagram. That is, aspects (such as layout, in this particular case) which, while not specified in the semantics of the language, may yet carry misleading implications. However, while the above work of Marks and Reiter illustrates the potential dangers of failing to take account for these two issues of selection of appropriate notations and of attention to secondary aspects, further research has indicated how the efficacy of diagrams for communicating information can be increased by actively exploiting these two issues.

In selecting a notation for presenting information one must take care that inherent properties of the notation do not carry misleading implicatures. However, where the implicatures inherent in a notation are well understood, they may be actively exploited by the semantics of the notation. This is achieved by assigning notations that carry inherent implicatures which capture and mimic important properties in the world which is being represented. For example, the diagram of Figure 7 carries implicatures due to the ordering by size of the node symbols. As these implicatures portray an ordering which does not exist in the world being represented, the network model, the diagram is misleading. Consider however a situation where there *does* exist an ordering between different kinds of nodes in the network model, and where this ordering is considered an important aspect of the model. In such a situation the choice of symbols which are similarly ordered by size, such as the circles in Figure 7, is ideal. In such a situation the semantics is carried by a universal implication and so the effort required in interpreting it is minimal.

This issue of selecting appropriate notations is an issue in the design of representational formalisms, of diagrammatic notations in this case. Naturally, criteria for determining the appropriateness of a notation require that the implicatures carried by notations (both those which are inherent in the graphics and those which are cognitive in nature—that is, depending upon human interpretation of graphical features) are well understood. A general framework in which to study this issue is given in Gurr (1996), which examines what it means for a diagram (or, more generally, any representation) to succeed or fail in being an isomorphism of that which it represents. Within this framework are discussed techniques for achieving isomorphism (that is, ensuring that the implicatures of a representation precisely match that which it represents). This issue of equivalence or isomorphism of diagram to that which it represents is similarly the subject of the theory of "specificity" (Stenning and Oberlander, 1995), which addresses the efficacy of diagrams in assisting human reasoning tasks. This theory suggests that it is the inherent implicatures and constraints of diagrams, which will generally limit their expressiveness, that make them so effective in supporting human reasoning. Finally, the inherent properties of various differing diagrammatic relationships are discussed in Zhang and Norman (1994). This paper, in the context of evaluating distributed cognitive tasks, introduces a taxonomy of various inherent properties of diagrammatic relations, permitting for example distinctions between continuous properties (such as relative size) and discrete properties (such as spatial layout).

The second issue illustrated by the previous diagrams is, as indicated in Figure 6, that secondary aspects of a diagram, although not specified in the semantics of the language, may yet carry misleading implications. However, studies have shown that in practice such secondary notations may be actively exploited by experienced users of diagrammatic languages. For example, a study of

digital electronics engineers using a CAD system for designing the layout of computer circuits demonstrated that the most significant difference between novices and experts was in the use of secondary notations, specifically layout information, in capturing information (Petre and Green, 1992). In such circuit diagrams the layout of components is not specified in the semantics as being meaningful. Nevertheless, experienced designers would exploit layout to carry important information by grouping together those components which were functionally related. By contrast, certain diagrams produced by novices were considered poor because they failed to utilise such secondary notations or, in particularly "awful" examples, were especially confusing through their mis-use of the common layout conventions adopted by the experienced engineers. This study and others are discussed in detail in recent work by Petre (1995), where the relationship between textual and graphical language and their respective benefits and limitations are presented. A major conclusion of this collection of studies is that the use of secondary notations is a significant contributory factor in the comprehensibility of graphical representations.

4 Applications: safety cases and standards

As noted in the introduction to this paper, a significant difference between safety critical systems and non-safety critical systems is in the industrial standards for safety critical systems. These standards often enforce mandatory requirements upon the design, development, testing, assessment and certification of safety critical systems. For safety critical systems in which software plays a significant role a newly emerging standard of particular import is IEC 1508 (IEC, 1995), which is currently in draft form and addresses the functional safety of such systems. According to this standard it is significantly harder to achieve, and demonstrate, the required levels of safety for AI software than it is for conventional software. The primary consequence of this conclusion is that IEC 1508 strongly recommends that AI techniques should not be used in anything but the least demanding of safety related applications.

Given the importance of standards the field of safety critical systems, and in particular the likely future importance of IEC 1508, it would appear that in the current industrial climate it is highly unlikely that KBS' will be acceptable in *critical* areas of safety-critical systems. This does not however exclude the use of KBS techniques for information presentation in, for example, user interfaces for safety critical systems, which are generally perceived as non-critical (although it is arguable that this is not so). In this section we discuss two further areas of safety-critical systems where KBS techniques for information presentation could be employed to good effect: in the construction of arguments of system safety and in the use of industrial standards.

A significant part of the design process for safety-critical applications is the construction of a "safety case", a collection of documents and data which together present the arguments for believing that the design of the proposed system is acceptably safe. The safety case sets out the risks involved with the operation of the process or equipment, and the possible consequences of failure. It also specifies what will be done (or has been done) to minimise the probability and the impact of these failures. The growing use of programmable devices is making system safety assessment increasingly complex, at the same time rendering many standard safety engineering techniques inapplicable. More and more frequently, safety cases have to combine the uncertainty of statistics with the abstract absoluteness of logics, and assess the validity of formal proofs in the context of unexpected and asynchronous events.

The safety case is a point of contact between experts from disciplines as diverse as electronic engineering, town planning, nuclear science and medicine. It contains contributions from a broad and distributed team of designers, suppliers and analysts. For large and complex systems, the production of the safety case may be the responsibility of the procurers or end users, who must therefore collate, organise and present evidence and arguments for safety supplied from both within their own organisation and from external suppliers. Safety cases are generally assessed by independent organisations who must ensure both that the arguments for safety are valid, and that relevant industrial standards and guidelines have been adhered to throughout. The safety case is

thus a collection of data drawn from multiple sources, representing diverse disciplines, which must be assessed by a range of both technical and non-technical experts, who exhibit a range of differing interests.

Given that a safety case is a diverse collection of assorted forms of data, and that it is often precisely the point of contact between the various parties involved in the production and assessment of safety critical systems, the communication of the information contained in a safety case is obviously an issue of primary importance. The discussions presented so far in this paper are therefore directly relevant to the structuring and presentation of safety cases. Consequently there is a very strong argument for the use of KBS techniques to fulfil this role for safety critical systems, both in structuring the information as the safety case is constructed and in tailoring the presentation of the safety case to its various readers.

For KBS techniques to be used in structuring information contained in a safety case it would be necessary to first identify the overall structure of the underlying argument of system safety which the safety case is intended to present. A study of safety-cases from a sociological perspective by MacKenzie (1996) suggests that, essentially, computer-system safety cases draw upon three kinds of argument:

- *construction*—the system is safe because the process used to design it is safe process that leads to safe systems;
- *induction*—this system is safe because testing shows that it is safe;
- *deduction*—this system is safe because it can be proven mathematically that its design is a correct implementation of its specification.

However, the acceptance of each of these three kinds of argument, and thus their usage in constructing a safety case, is strongly influenced by a more sociological matter, that of *authority*. Matters of authority can shape the processes of construction induction and deduction. Indeed an argument of safety by any of the above three means involves ultimately some appeal to authority on the part of the party supplying the argument, and acceptance of the argument involves acceptance of this authority on the part of the reader. That is, any argument ultimately will involve an appeal that, say, a particular development method is safe; a certain testing regime is generally accepted as a sufficient indicator of safety; or that a formal model of some implemented system is an accurate and sufficient abstraction of it.

For safety critical computer systems one of the most potent authorities which can be, and is, appealed to is a relevant standard, such as IEC 1508. This standard, in common with many of the standards concerning software related safety critical systems, is based around a particular lifecycle model of computer system development. For each phase of the lifecycle are set out requirements for documentation which specifies the state of the design at the commencement of that phase and the safety requirements which must be demonstrated as being satisfied by the end of that phase. Guidelines are given both for techniques to achieve the completion of each phase and for techniques to assess the satisfaction of the necessary safety requirements upon completion. Many other relevant standards, for example MOD Defence Standards 00-55 (MOD, 1993a) and 00-56 (1993b), follow broadly similar lines; giving guidelines for assessment of satisfaction of safety requirements based around some lifecycle model of the process which the standard addresses (e.g., software procurement and hazard analysis, respectively, in the case of MOD 00-55 and MOD 00-56).

A safety case for a computer-system, built according to the lifecycle model suggested by IEC 1508, would likely refer to this standard. The safety case would demonstrate that, at each phase, techniques for proving safety, both inductive and deductive, suggested by the standard have been satisfactorily applied. This argument (which involves arguments of construction, induction and deduction) is effectively based primarily on an appeal to the authority of the standard. This implies two things. First that such standards indicate the form and format of arguments for the safety of systems (computer-related and otherwise) which are commonly accepted by industry. Consequently such standards from a useful focus of attention for any knowledge engineer who wishes to become familiar with the types of arguments commonly seen in safety-cases. The second implication is that

these standards would themselves benefit substantially from the support of KBS techniques for information presentation, for communicating their arguments and underlying assumptions.

5 Review: KBS techniques for information presentation

In this section we present a review of KBS techniques and existing systems from the perspective of information presentation utilising multiple multi-media representations. We review first frameworks for information presentation, moving on through various single-media representations and ultimately to multi-media information presentation.

5.1 Information presentation: theory and frameworks

In many applications the type of information to be presented to the user is known to be relatively straightforward it may be predetermined and presented via a relatively simple interface. This may hold even of systems which exhibit a sophisticated response to user needs, such as on-line help systems. In the on-line help system of Harley and Smith (1988) for the EUROHELP project (Breuker, 1990) for example, interactions with the user are of a known and straightforward nature. Here interaction with the user is via menus of questions generated by the system based upon knowledge of the user's current task and current knowledge state. For the presentation of formal descriptions of safety critical systems, given the ranges of the natures, modelling techniques, users and assessors of such systems, a more sophisticated approach to the presentation of information is necessary. Knowledge based systems have been employed with some success to this end in information presentation systems (Murray, 1994).

Information Presentation Systems (IPS) automatically generate graphical or multi-mediapresentations of data. Typically they are KBS that use knowledge of graphic design, presentation styles and target domain. The basis for many current IPS was that provided by the APT (''A Presentation Tool'') generic system of MacKinlay (1986). This established that combinations of conventional presentation graphics can be designed automatically using domain-independent graphical description languages. MacKinlay suggested a categorisation of this and future information presentation systems according to three criteria: content selection; graphic design issues; and the range of graphic designs available to system. For the presentation of formal specifications or arguments we both extend and interpret this categorisation to suggest the three key issues of presentation:

- content selection: from a formal description, and potentially a model of the user, what information should be presented;
- range of languages (textual, graphical, audio, video, etc.) and architectures (single media, multimedia, hypertext, etc.);
- multimedia design issues: allocation of information to media, layout, structuring of hypertext (if available).

The fundamental generic approach of APT was considerably extended by the SAGE (Roth and Mattis, 1990) and later the BOZ (Casner, 1990) systems. In SAGE informational goals were used to govern selection of the appropriate graphical presentation while BOZ used task goals of what the graphic is for the same purpose. SAGE is an explanation system that uses a combination of text and graphics to explain changes that occur in quantitative modelling systems such as spreadsheets. The intention of SAGE was more general, being a system composed of components that are ''as generic and application independent as possible''.

Alternative frameworks are, of course, possible. One particularly attractive, theoretically motivated, modality independent framework for describing, organising and producing multimedia discourse is that of Stein and Maier (1995). This applies theories of human communication, in particular speech act theory and Rhetorical Structure Theory (RST) (Mann and Thompson, 1987, 1988; Moore and Pollack, 1992), in a framework where all user and system actions are represented as communicative acts. These acts are categorised by purpose and then organised into

dialogue scripts. Attention has also recently focussed on alternatives to traditional top-down hierarchical planners. One example being Hovy and Arens (1993), who argue that this traditional approach is inadequate in general and suggest the use of several interacting reactive planners required.

In contrast to the generic approach of APT and its descendants, systems such as CUBRICON system of Neal and Shapiro (1991) and the APEX system of Feiner (1985) exploited application specific information to structure the graphical presentation of information and provide a range of sophisticated features. The Playfair system of Marshall (1993) supports the user in selecting suitable graphical presentations of business graphics (line charts, scatter plots, etc.). While not precisely application specific, by restricting the range of graphical languages available Playfair may similarly exploit knowledge of the expected type of information to be presented. Playfair combines design templates with a rule-based system for evaluating suitability of the selected templates for a particular data set and includes of the user in the presentation selection cycle. This is a considerably different approach to both those of APT and APEX and reflects trends in other areas to provide support to designers rather than replacing them in the design process.

5.2 Natural Language Generation

As has already been emphasised, the use of natural language to present formal descriptions is a necessary part of any generic approach. While such presentations at present are generally performed manually, the surge of interest of the past decade in the automatic generation of sizable passages (''paragraph length'') of natural language text has led to a number of systems which could be employed to great benefit to automate the presentation of formal descriptions. Similar suggestions have already been made, and in some cases employed, for KBS applications, such as user-critiquing (Rankin, 1993).

Natural language generation is commonly viewed, although not exclusively so, as being in two stages: deep generation and surface generation. In deep generation the desired content of the text is determined. At this stage decisions are made as to what should be included and how best this content should be organised. In surface generation the appropriate words, phrases and syntactic structures are selected so as to produce the required message. Current research on surface generation (Dale et al., 1990, 1992) deals with issues such as lexical choice, word order, how to realise syntactically information about topic/comment and focus/background.

The past decade has seen a growing interest in generating paragraph length (as opposed to single sentence) passages of text. This has required advancements in deep generation issues, as generating such longer texts requires a more sophisticated understanding of the basic units of text. To generate long passages of text it is necessary to understand what are the units in a text, how units are to be ordered and how units in the text are related to one another and to the text as a whole.

Early influential work which unified work on focus of attention, discourse or linguistic structure, and discourse intentions was by Grosz and Sidner (1986). At a similar time McKeown (1985) first introduced the notion of using schemas of rhetorical predicates in the generation of texts. A schema ''may be viewed as representing a strategy typically used to meet a particular communicative goal and provide the means for organising the predicates into a coherent text''. Such schemas have proved effective, as in the TEXT system of Miller and Rennels (1988), which used schemas of rhetorical predicates to produce paragraph-length texts as answers to queries about contents of a database. One drawback of this approach has been an explosion in the number of schemas necessary in order to generalise across domains, although work is still in progress in progress in reusing schemas for different discourse purposes to achieve such generalisation (McKeown et al., 1990).

An alternative approach to such a general theory of text organisation is to define a grammar for particular purpose. An example of such is the explanation grammar of Cawsey (1989).

More recently, there has also been work which has examined the relationships between stretches of text. These approaches generally follow the argument that texts are cohesive by virtue of the rhetorical and semantic relationships that hold between the units of text. Most notable of these

approaches is Rhetorical Structure Theory (RST) (Mann and Thompson, 1987, 1988; Moore and Pollack, 1992) which, as pointed out by Hovy (1990) , subsumes large parts of other work, including for example McKeown's schemas. RST has been used as basis for text generation, for example Hovy (1988), Moore and Swartout (1989) and Moore and Paris (1989). It has also been extended to cope with other media beyond natural language. For example by André and Rist (1990), who incorporate diagrams as satellites in explanatory text.

Natural language generation techniques have been further extended to deal with text generation in hypertext systems. Work in this area includes the IDAS system (Reiter et al., 1995) for producing technical documentation. In IDAS, text is derived from a single non-textual documentation database, rather than text being itself the primary representation used. This helps to make the documentation both adaptable and reusable. This approach has been extended yet further in the CORECT system (Levine and Mellish, 1994), a multi-user tool for the authoring of requirements specifications. CORECT is a system to help in the design of an electronic system in response to a call for tenders from a customer. The system is particularly interesting in its combinations of CSCW technology (Computer-Supported Cooperative Working) for multi-user input with natural language generation for output. In CORECT automatic generation of documents gives different users different views of the underlying knowledge pool, in response to the different information needs of the participants in the design process. The automatic generation of explanatory natural language descriptions in a hypertext environment is also the subject of the ILEX system (Knott et al., 1996). ILEX seeks to satisfy multiple objectives (addressing the level of expertise of the reader; relating generated text to a history of previously presented texts so as to avoid repetition; and satisfying educational goals) in producing textual descriptions of objects encountered during a guided tour of a "virtual" (initially) museum gallery.

5.3 Graphics and limited mixed media approaches

The approach in many early KBS interfaces adopted a relatively simplistic view of the efficacy of graphics, the common assumption being that any graphical interface was advantageous to the user. Such systems are typified by the VEGAN editor of Esfahani and Kellet (1988), which permitted the user to structure knowledge via associative networks. While such systems often were preferable to text only interfaces the use of graphics did not generally consider human cognitive needs and as such is open to the general criticisms of Petre (1995) discussed previously. Consideration to cognitive needs and the efficacy of graphics was present in some systems however, as evidenced in the review of early interfaces by Jones (1988).

Graphical interfaces for the presentation of any information are likely to be successful in any application area which has existing graphical languages. Where these are in common use they may be exploited to great effect, as in the JANUS system of Fischer et al. (1989). JANUS supported kitchen design via a graphical interface which utilised the standard graphical languages employed by architects and planners. This permitted users to plan kitchen layouts using familiar icons and tokens. In a similar manner the AMPHION system of Lowry et al. (1994) presents a very nice example of the domain-oriented knowledge engineering approach (Fischer, 1992), where the structure of the interface is closely morphic to the application domain. AMPHION is intended as a generic architecture which has so far been instantiated to a particular system which assists in the development of NASA software for performing astronomical calculations. The AMPHION interface is graphical (specifically, geometrical) in nature and programs (combinations of library functions) are specified via solar system kinematics diagrams.

A number of researchers have sought to exploit the potential of graphical languages to focus attention in dialogues and thus simplify communication. The basic idea of much of this work has been to establish 'reified dialogues' between system and user, where as much information as possible is communicated via the modification of displayed graphics rather than by generation of texts. This includes the human knowledge engineering systems of Terveen (1993) and Terveen et al. (1991) which compute various kinds of critiques as a user adds and links object in knowledge diagram. The

systems critiquing is communicated via colour shading and variations in fonts within the diagrams and thus illustrates a particularly attractive use of secondary notations in communicating information. The DETENTE system of Wroblewski et al. (1991), which partially automates task management, uses similar display techniques to communicate task status. Other similar systems utilise reified dialogues in critiquing software construction (Terveen and Selfridge, 1994) and the construction of Petri-nets (Stolze, 1994).

5.4 Multi media approaches

The most successful use of multimedia presentations of information in KBS interfaces have been in application specific rather then general purpose systems. APEX (Feiner, 1985, 1991), and later COMET (Feiner and McKeown, 1990), is the main example of early work. These systems utilise a frame based database of objects together with a set of rules to determine how much detail is needed for each frame. The display of information is limited only to that which is needed and this permits a quite sophisticated combination of text and graphics in information presentation.

Other notable application specific systems are CUBRICON and Integrated Interfaces. CUBRICON (Neal et al., 1988; Neal and Shapiro, 1991) is a military presentation system which utilises a wide range of media including colour graphics, tables, text and spoken language. It employs an extensive knowledge base of of application specific data together with a detailed set of task descriptions. In response to user queries CUBRICON selects a relevant subset of available data and suitable presentation media. The selection of media itself determines a limited number of presentation formats from which a selection is made according to guidelines based on information type. Integrated Interfaces (Arens et al., 1988, 1991) is a set of tools integrated around a knowledge base for modelling naval ship movements. Models of the application domain, interface and a final model of functions and data structures of available applications all integrated into the knowledge-based and utilised in information presentation. Presentation media available include natural language, graphics (including maps), tables and forms.

While earlier generic systems such as APT and BOZ concentrated upon information layout within a single media, the coordination and layout of information in a multimedia presentation has recently received greater attention. The majority of notable approaches employ theoretical frameworks of human communication to structure multi-media presentations. Examples include the use of communicative acts by Maybury (1991) and the speech-theoretic approach taken by the WIP system, of Rist et al. (1993). WIP considers the generation of a multi-media document as an act sequence which aims to achieve certain goals. Recently, Graf (1995) has proposed a framework for WIP specifically to address multimedia layout. This framework represents three types of communicative acts in an independent manner and uses a planner to coordinate textual and graphical displays.

Of final note for layout of multi-media information is the COMET system (Feiner and McKeown, 1990) mentioned previously. COMET uses interactively generated text and 3D graphics to explain repair and maintenance tasks and uses "communicative goals" to govern presentation of graphics. While COMET is application specific it takes a very interesting approach to the planning of information presentation which could well generalise. In COMET a content planner produces representation of information to be communicated in a media independent Logical Form (LF). A media coordinator annotates LF to specify those parts which may be presented as text, as graphics, or as a combination of the two. Two independent generators are used for text and graphics respectively and these generators themselves may coordinate by further LF annotations.

6 Conclusions

Given the status of industrial standards in the field of safety critical systems, and the attitude towards AI software portrayed by the (potentially highly influential) IEC 1508 standard, it would seem that KBS are unlikely to be accepted for use in any aspect of a safety critical system which is

considered to be critical. This does not, however, imply that there is no place for KBS' in safety critical systems. It is for the structuring and presentation of information that KBS techniques can, and should, be used to good effect. This includes not only user interfaces for safety critical systems but also industrial standards and safety cases, the arguments for system safety which are a required part of the design of any safety critical system.

The structuring and presentation of information for communicative purposes is an issue of primary importance in the development and assessment of safety critical systems. This is due to the nature of the design and assessment of such systems, which often involves distributed teams of designers, suppliers, users and analysts who represent diverse areas of expertise and motivations and a broad range of technical and non-technical disciplines. The use of formal descriptions can significantly assist in ensuring the correctness and consistency of information throughout the design and assessment process. However, formal descriptions are of little benefit unless they are made accessible to the diverse audiences involved in the development and assessment of safety critical systems. In this paper, we have argued that the major task required to make formal descriptions accessible is the effective communication of the assumptions which underly them. The effect of communicating underlying assumptions is twofold. Firstly it permits the validation of a formal description. Secondly it provides a sense of context which assists a reader in correctly interpreting the consequences of a given formal description.

We have argued further in this paper concerning the need for multiple multimedia presentations of information, specifically of the assumptions underlying formal descriptions. It is for this purpose that, we have proposed, KBS techniques can be used to greatest effect: in the selection and structuring of information, its allocation to particular media, the coordination of different representations, and user interaction with the presented information. The review of KBS systems and techniques presented in section 5 has been structured with this purpose in mind. However, we have sought also to demonstrate that when communicating information to a human reader, particularly in safety critical applications where it is essential that communication be both unambiguous and effective, it is important that communication be based upon a sound theoretical foundation. We have shown by example that to guide the allocation of information to different media, and to understand the impact (which is partly cognitive) of these media upon a human reader, such theories are vital. Theories of human communication, which must also acknowledge the role of cognitive issues in interpretation, offer benefits to information presentation systems by providing a rational basis for the selection and coordination of notations which are effective both through avoiding being misleading or ambiguous and through actively exploiting the communicative properties of various media.

Acknowledgements

The author wishes to thank the fellow authors of this issue of *The Knowledge Engineering Review* for their helpful and insightful comments regarding the contents and, most particularly, the structure of this paper. This work was supported by EPSRC/DTI Safety Critical Systems project GR/J58619, "Communication in Safety Cases: A Semantics Approach".

Further reading

The following are recommended as further reading for an overview of the numerous fields of KBS research relevant to the communication of information and explanations:

Information Presentation Systems: Murray (1994) provides an overview of the two major development paths (generic and application specific) for information presentation systems. The article reviews early systems, their problems and shortcomings, and also later systems through to state of the art. It concludes with a presentation of outstanding research issues.

Explainable Expert Systems: this is a framework for building expert systems capable of explaining their reasoning as well as their domain knowledge (Swartout et al., 1992).

Intelligent User Interfaces: one of the first discussions on general features of intelligent interfaces was that provided by Rissland (1984). A workshop was held at Monterey, California in 1988 on "Architectures for Intelligent Interfaces: Elements and Prototypes" and a subset of the papers presented there are collected in Sullivan and Tyler (1991). Finally, an overview and bibliography of "human interaction with intelligent systems" is provided by Woods et al. (1991).

Human Computer Collaboration: thirty researchers came together in October 1993 for a AAAI Fall Symposium dedicated to Human Computer Collaboration (HCC). Terveen (1995), in a special issue of the *International Journal of Knowledge Based Systems* (Volume 8, No. 2-3, December 1993) dedicated to the results of that symposium, provides an overview of HCC and derives a set of fundamental issues. The two major approaches (AI and HCI) are introduced and surveyed with respect to these issues. Finally Terveen proposes themes with which a unified approach may be achieved.

Intelligent Help: Carroll and McKendree's (1987) foundational paper presents requirements for advice-giving systems and a taxonomy of some of the key issues in the design of such systems. A comprehensive reference for research in problems of user applications and of building adequate help facilities is given in the concluding report from the EUROHELP project (Breuker, 1990). This report also documents the development methodology used for designing and developing intelligent help systems.

Critiquing: a recent issue of the *Knowledge Engineering Review* (Volume 8, No. 4, December 1993) is dedicated to this field. An example of a critiquing system which employs a graphical interface to good effect is the JANUS system of Fischer *et al*. (1989), which supports kitchen design via a graphical interface. Designs were critiqued by JANUS which employed an issue-based explanation system to provide feedback to the user. This permitted the user to explore the arguments for particular critiquing messages. The explanations provided could be based either on differences between user and system solutions (differential critiquing) or on violations of general design rules (analytical critiquing).

On-Line Assistance: Wasson (1992) provides an overview of On-Line Assistance (OLA) research: from empirical studies of effectiveness to the design of intelligent OLA's capable of tailoring assistance to individual users. The article concludes that there are numerous empirical question still to be answered. For instance: What types of assistance best for what types of user? What kind of advisory strategies are there? Under what conditions are different strategies effective? How should assistance be accessed and displayed? This paper concludes with a bibliography of introductory reading in a number of related fields.

References

IEC 1508, 1995, *Draft International Standard 1508 "Functional Safety: Safety Related Systems"*. (Available from British Standards Institution, BSI Sales Office, 389 Chiswick High Road, London W4 4AL.)

Arens, Y, Miller, L and Sondheimer N, 1991, "Presentation design using an integrated knowledge base" in JW Sullivan and SW Tyler (eds), *Intelligent User Interfaces* 241–258. ACM Press.

Arens, Y, Miller, L, Shapiro, S and Sondheimer, N, 1988, "Automatic construction of user-interface displays" *Proc. AAAI'88* 808–813.

André, E and Rist, T, 1990, "Towards a plan-based synthesis of illustrated documents" *Proc. ECAI'90* Stockholm.

Barwise, J and Etchemendy, J, 1994, *Hyperproof* CSLI Publications.

Bowen, J and Stavridou, V, 1993, "The industrial take-up of formal methods in safety-critical and other areas: a perspective" in JCP Woodcock and PG Larsen (eds), *FME'93: Industrial-Strength Formal Methods*, volume 670 of *Lecture Notes in Computer Science* 183–195. Springer-Verlag.

Bowen, J and Stavridou, V, 1993 "Safety-critical systems, formal methods and standards" *Software Engineering Journal*, July.

Breuker, J (ed.), 1990, "EUROHELP: Developing intelligent help systems" Technical report, Kopenhagen, Amsterdam/Manchester, Leeds, EC.

Casner, SM, 1990, "A task-analytic approach to the automated design of information graphics" PhD thesis, University of Pittsburgh.

Cawsey, A, 1989, "Explanatory dialogues" *Interacting with Computers* **1**(1) 69–92.

Carroll, JM and McKendree, J, 1987 "Interface design issues for advice-giving expert systems" *Communications of the ACM* **30** 14–31.

Chemical Industries Association, 1992, "A Guide to Hazard and Operability Studies" HAZOPS Guide.

Cox, R, Stenning, K and Oberlander, J, 1994, "Graphical effects in learning logic: reasoning, representation and individual differences" *Proc. 16th Annual Meeting of the Cognitive Science Society*.

Dale, R, Hovy, E, Rösner, D and Stock, O (eds), 1992, *Aspects of Automated Natural Language Generation: Proceedings 6th International Workshop on Natural Language Generation*.

Dale, R, Mellish, C and Zock, M (eds), 1990, *Current Research in Natural Language Generation* Academic Press.

Esfahani, L and Kellett, J, 1988, "Integrated graphical approach to knowledge representation and acquisition" *Knowledge-Based Systems* **1**(5) December.

Feiner, S, 1985, "APEX: An experiment in the automated creation of pictorial explanations" *IEEE Computer Graphics Applications* **5**(11) 29–37.

Feiner, S, 1991, "An architecture for knowledge-based graphical interfaces" in JW Sullivan and SW Tyler (eds) *Intelligent User Interfaces*, 259–279. ACM Press.

Feiner, S and McKeown, KR, 1990, "Coordinating text and graphics in explanation generation" *Proc. AAAI'90* 442–449, Boston, MA.

Fischer, G, 1992, "Domain oriented design environments" *Proc. 7th KBSE* 204–213.

Fischer, G, McCall, R and Morch, A, 1989, "JANUS: integrating hypertext with a knowledge-based design environment" *Proc. Hypertext'89* 105–117.

Graf, WH, 1995, "Intent-based layout in interactive multimedia communication" in J Lee (ed.) *First International Workshop on Intelligence and Multimodality in Multimedia Interfaces: Research and Applications* Edinburgh, UK.

Grice, H, 1975, "Logic and conversation" in P Cole and J Morgan, (eds) *Syntax and Semantics: Vol 3, Speech Acts* 43–58. Academic Press.

Grosz, BJ and Sidner, CL, 1986, "Attention, intentions, and the structure of discourse" *Computational Linguistics* **123** 175–204.

Gurr, C, 1995, "Supporting formal reasoning for safety critical systems" *High Integrity Systems* **1**(4) 385–396.

Gurr, C, 1996, "On the isomorphism (or otherwise) of representations" *International Workshop on Theory of Visual Languages, held in Conjunction with AVI'96* Gubbio, Italy.

Hovy, E and Arens, Y, 1993, "The planning paradigm required for automated multimedia presentation planning" *Human-Computer Collaboration: Reconciling Theory, Sythesizing Practice. AAAI 1993 Fall Symposium* AAAI Tech report FS-93-05.

Hall, A, 1990, "Seven myths of formal methods" *IEEE Software* September.

Hovy, E, 1988, "Approaches to the planning of coherent text" *Proc. 4th International Workshop on Natural Language Generation* Los Angeles, CA.

Hovy, E, 1990, "Unresolved issues in paragraph planning" in R Dale, C Mellish, and M Zock (eds) *Current Research in Natural Language Generation*. Academic Press.

Harel, D, Pnueli, A, Schmidt, JP, and Sherman, R, 1986, "On the formal semantics of statecharts" in *Proc. First IEEE Symp. Logic in Computer Science*.

Harley, JR and Smith, MJ, 1988, "Question answering and explanation giving in on-line help systems" in J Self (ed.) *Artificial Intelligence and Human Learning* 338–360. Chapman & Hall.

Jones, S, 1988, "Graphical interfaces for knowledge engineering: An overview of relevant literature" *Knowledge Engineering Review* **3**(3) September.

Knott, A, Mellish, C, Oberlander, J and O'Donnell, M, 1996, "Sources of flexibility in dynamic hypertext generation" *Eighth International Workshop on Natural Language Generation* 151–160, Sussex, UK.

Levine, J and Mellish, C, 1994 "CORECT: Combining CSCW with natural language generation for collaborative requirements capture" *Seventh International Workshop on Natural Language Generation* 236–239, Kennebunkport, Maine.

Lowry, M, Philpot, A, Pressburger, T and Underwood, I, 1994, "A formal approach to domain-oriented software design environments" *KBSE'94*.

Mackinlay, J, 1986, "Automating the design of graphical presentations of relational information" *ACM Transactions on Graphics* **5**(2) 110–141.

MacKenzie, D, 1996, "A Worm in the Bud? Computers, Systems, and the Safety-Case Problem" *Proc. Dibner Institute Symposium on "The Spread of the Systems Approach"* Cambridge, MA.

Marshall, R, 1993, "Improving the quality of computer generated charts" in Salvendy and Smith (eds) *Human-Computer Interaction: Software and Hardware Interfaces – HCI'93* 597–602. Elsevier.

Maybury, MT, 1991, "Planning multimedia explanations using communicative acts" *Proceedings AAAI'91* 61–66.

McKeown, K, 1985, *Text Generation* Cambridge University Press.

McKeown, K, Elhadad, M, Fukumoto, Y, Lim, J, Lombardi, C, Robin, J and Smadja F, 1990, "Language generation in COMET" in R Dale, C Mellish and M Zock (eds) *Current Research in Natural Language Generation*. Academic Press.

Milner, R, 1989, *Communication and Concurrency* Prentice Hall.

MOD 00-55, 1993, *The Procurement of Safety Critical Software in Defence Equipment*. *Part 1: Requirements & Part 2: Guidance*. Ministry of Defence Interim Standard.

MOD 00-56, 1993, *Hazard Analysis and Safety Classification of the Computer and Programmable Electronic System Elements of Defence Equipment*. Ministry of Defence Interim Standard.

Moore, J and Paris, C, 1989, "Planning text for advisory dialogues" *Proc. 27th Annual Meeting of the Association for Computational Linguistics* Vancouver Canada.

Moore, JD and Pollack, ME, 1992, "A problem for RST: The need for multi-level discourse analysis" *Computational Linguistics* **18**(4) 537–544.

Miller, P and Rennels, G, 1988, "Prose generation from expert systems" *AI Magazine* **9**(3) 37–44.

Marks, J and Reiter, E, 1990, "Avoiding unwanted conversational implicature in text and graphics" *Proc. AAAI-90* 450–456.

Moore, J and Swartout, W, 1989, "A reactive approach to explanation" *Proc. IJCAI'89* 1504–1510.

Mann, W and Thompson, S, 1987, "Rhetorical structure theory: Description and construction of text structures" in G Kempen (ed.) *Natural Language Generation* Martinus Nijhoff.

Mann, W and Thompson, S, 1988, "Rhetorical structure theory: toward a functional theory of text organization" *Text* **8** 243–281.

Murray, BS, 1994, "Intelligent information presentation systems" *Knowledge Engineering Review* **9**(3) 269–286.

Neal, JG, Bettinger, KE, Byoun, JS, Dobes, Z and Theilman, CY, 1988, "An intelligent multi-media human-computer dialogue system" *Proc. Workshop Space Operations, Automation and Robotics (SOAR'88)* 245–251, Wright State University.

Neal, JG and Shapiro, SC, 1991, "Intelligent multi-media interface technology" in JW Sullivan and SW Tyler (eds) *Intelligent User Interfaces* 11–43. ACM Press.

Oberlander, J, 1996, "Grice for graphics: pragmatic implicature in network diagrams" *Information Design Journal* **8**(2) 163–179.

Petre, M, 1995, "Why looking isn't always seeing: Readership skills and graphical programming" *Communications of the ACM* **38**(6) 33–45, June.

Petre, M and Green, TRG, 1992, "Requirements of graphical notations for professional users: electronics CAD systems as a case study" *Le Travail Humain* **55** 47–70.

Pnueli, A and Shalev, M, 1992, "What is in a Step: On the Semantics of Statecharts" *TACS* Springer-Verlag, LNCS 526.

Rist, T and André, E, 1993, "Designing coherent multi-media presentations" in Salvendy and Smith (eds) *Human-Computer Interaction: Software and Hardware Interfaces – HCI'93* 434–439. Elsevier.

Rankin, I, 1993, "Natural language generation in critiquing" *Knowledge Engineering Review* **8**(4) 329–347, December.

Rissland, EL, 1984, "Ingredients of intelligent user interfaces" *International Journal of Man-Machine Studies* **21** 377–388.

Roth, SF and Mattis, J, 1990, "Data characterization for intelligent graphics presentation" *Proc. CHI'90* 193–200. ACM.

Reiter, E, Mellish, C and Levine, J, 1995, "Automatic generation of technical documentation" *Applied Artificial Intelligence* **9**(3), 259–287.

Sullivan, JW and Tyler, SW (eds), 1991, *Intelligent User Interfaces* Adison-Wesley.

Stein, A and Maier, E, 1995, "Structuring collaborative information-seeking dialogues" *Knowledge Based Systems* **8**(2) 82–93.

Stenning, K and Oberlander, J, 1995, "A cognitive theory of graphical and linguistic reasoning: logic and implementation" *Cognitive Science* **19** 97–140.

Swartout, WR, Paris, CL and Moore, JD, 1992, "Design for explainable expert systems" *IEEE Expert* **6**(3) 58–64.

Stenning, K, 1995, "Logic as a foundation for a cognitive theory of modality allocation" L Polos and M Masuch (eds) *Applied Logic — How, What and Why?: Logical approaches to natural language* 321–342. Kluwer.

Stolze, M, 1994, "Visual critiquing in domain oriented software environments: showing the right thing at the right place" *Proc. AI and Design*.

Terveen, LG, 1993, "Intelligent systems as cooperative systems" *International Journal of Intelligent Systems* **3**(2–4) 217–250.

Terveen, LG, 1995, "Overview of human computer collaboration" *Knowledge-Based Systems* **8**(2–3) 67–81, April–June.

Terveen, LG and Selfridge, PG, 1994, "Intelligent assistance for software construction: a case study" *Proc. 9th KBSE*.

Terveen, LG, Wroblewski, DA and Tighe, SN, 1991, "Intelligent assistance through collaborative manipulation" *Proc. IJCAI'91* 9–14.

Wasson, B and Akselsen, S, 1992, "An overview of on-line assistance: from on-line documentation to intelligent help and training" *Knowledge Engineering Review* **7**(4) 289–322.

Wroblewski, DA, Hill, WC and McCandless, T, 1991, "DETENTE: practical support for practical action" *Human Factors in Computer Systems: Proceedings CHI'91* 195–202.

Woods, DD, Johannesen, L and Potter, SS, 1991, "Human interaction with intelligent systems: An overview and bibliography" *SIGART Bulletin* **2**(15) 39–50.

Zhang, J and Norman, D, 1994, "Representations in distributed cognitive tasks" *Cognitive Science* **18** 87–122.