Implementing Configurable Information Systems: A Combined Social Science and Cognitive Science Approach

Corin Gurr and Gillian Hardstone

IRC for Dependability of Computer Based Systems (DIRC),
University of Edinburgh, 2 Buccleuch Place, Edinburgh EH8 9LW, UK
`{C.Gurr, G.Hardstone}@ed.ac.uk`,
`http://www.dirc.org.uk`

Abstract. This paper outlines an interdisciplinary approach to tackling the issues of integrating medical information systems into existing healthcare environments where high dependability is a significant requirement. It focuses on the knowledge of system users (domain practitioners) and designers, and the potential use of diagrammatic representations of that knowledge during the implementation process in order to support communication between the two groups, and to serve as tools in assisting system reconfiguration to user requirements during implementation.

1 Introduction and Background

This paper outlines an interdisciplinary approach to tackling the issues of integrating medical information systems into existing healthcare environments where high dependability is a significant requirement. It focuses on the knowledge of system users (domain practitioners) and designers, and the potential use of diagrammatic representations of that knowledge during the implementation process in order to support communication between the two groups, and to serve as tools in assisting system reconfiguration to user requirements during implementation.

Integration of new technological systems into an existing organisational environment requires a clear understanding of technology as intrinsically social [14], rather than as predominantly technical, but with social aspects. This makes it easier to unravel some of the implications of implementing a technology in a particular environment, including changes in processes; shifts in power relations, responsibility, authority and access to information; and how these factors interact.

Knowledge is an important aspect of technology [14], and thus a key issue in system design and implementation. It is intrinsically social, both in terms of its substantive content (what is known) and its cognitive content (how it is known). New systems need to interface with users' existing knowledge of the domain(s) in which they operate, and the activities (practice) that need to be performed within the domain space in an organisational context. Designers also need knowledge about users' domains of knowledge and the specific context in which they put that knowledge into practice in order to communicate effectively during elicitation and requirements analysis. The relation between designers and users is critical when implementation involves extensive system reconfiguration [6] to user needs, as is usually the case with Hospital Information Systems (HIS).

The use of diagrammatic representations is common throughout engineering and design practice [8]. Adopting a social science-based approach to knowledge and practice in organisations, we intend to work from ethnographic descriptions of knowledge and practice in a specific empirical healthcare setting, as informed by taxonomies of knowledge [5, 11, 20], through to a representation of that knowledge in diagrammatic forms, in order to facilitate communication between users and designers. We anticipate that our methods will result in the development of a useful tool for systems implementation.

2 Knowledge and Practice in Organisations

A significant issue for the design and implementation of IT systems which are intended to support the business or operating processes of complex organisations, is how the deployment of these systems actually changes those processes, intentionally or otherwise. Integration of technological systems into an existing organisational environment requires a clear and visible understanding of the potential ramifications of the technology in that particular environment: which processes or practices will or could change; how responsibility, authority and access to information may change and how these three factors interact.

One approach to the issues outlined above is to consider knowledge as a key factor in computer system design and implementation, particularly when a new system is being designed to replace an existing system. System designers need to know about users' domains of knowledge: the content of that knowledge, how it is structured, and how it is used. To acquire this knowledge, they need to communicate effectively with potential system users to elicit and analyse requirements. Most importantly, the new system needs to interface with users' existing knowledge of the domain(s) in which they operate, and the activities that need to be performed within the domain space. Users also need to understand how to use the new system in an organisational context once it has been designed. The relation between designers and users is critically important if the implementation process involves extensive customisation or reconfiguration of a basic system to user needs, when design and innovation continue during the system's operation within the user organisation, as is often the case with Hospital Information Systems (HIS).

But how to understand and capture the complexity of organisational knowledge and practice during the system design and configuration process? And how to convey that knowledge between designers and users? Exploring who knows what in the domain space, and what they do with their knowledge is a useful point of entry into this area. A sociologically-influenced approach to domain knowledge is proposed.

Knowledge can be seen as an inherently social process, in terms of its cognitive and substantive content, distribution and mobilisation for practice [12]. There is an existing body of work, primarily in the sociology of science and technology, that deals with different ways of categorizing knowledge from a social science perspective. For example, Vincenti [20] has analysed the substantive components of domain knowledge, relating them to the knowledge-generating activities that create them, in the domain of aeronautical design engineering (see Table 1). This framework thus appears domain-specific, but can readily be adapted to other domains [11]. Although Vincenti's work does not specifically address the social nature of knowledge, it implies a division of labour within a given domain: different people will be carrying out various activities (such as research, experimental work or operation), and the distribution of the substantive content of knowledge will therefore vary accordingly, and be unequal. Such considerations clearly have implications for the implementation of a HIS, where there are many and varied occupational groups of users, including administration staff and clinicians from a range of domains and sub-domains.

Table 1. Categories of substantive (aeronautical design engineering) knowledge, and knowledge-generating activities [20].

Categories **Activities**	Fundamental design concepts	Criteria and specifications	Theoretical tools	Quantitative data	Practical considerations	Design instrumentalities
Transfer from science			X	X		
Invention	X	X	X	X		X
Theoretical research	X	X	X	X		X
Experimental research	X					
Design practice		X			X	X
Production				X	X	X
Direct trial (including operation)	X	X	X	X	X	X

The cognitive content of knowledge is no less socially shaped and distributed. One taxonomy that captures these aspects of cognition is that developed by Fleck [5, 7] (see Table 2). It goes beyond the conventional distinction between tacit and explicit knowledge, and may be useful in a systems design context, because it carries considerable explanatory power about social relations and context. For example, meta-knowledge about a domain is likely to be shared by most people working at a site, or forming part of the same department or occupational group. Although formal knowledge is often highly valued (just one reason for the status of clinicians) and rewarded, in a workplace context, it is often not the most important or useful for everyday practice. For example, contingent, locally-specific knowledge is usually extremely important during the implementation of configurational technologies, such as HIS, but often undervalued.

Table 2. Components and contexts of knowledge (After Fleck [7]).

Components of knowledge

Cognitive components	**Description**	**Acquired through**	**Embodied in**
Formal knowledge	Theories, formulae; often in written or diagrammatic form	Formal education	Codified theories
Informal knowledge	Rules of thumb, tricks of the trade	Interaction within a specific milieu	Verbal interaction
Contingent knowledge	Widely distributed, seemingly trivial information, context-specific	On the spot learning	The specific context
Tacit knowledge	Rooted in practice and experience	Apprenticeship and training	People
Instrumentalities	Embodied in the use of tools or instruments	Demonstration and practice	Use of tools
Meta-knowledge	General cultural and philosophical assumptions; values, goals; may be specific to organisation, domain, occupational group, etc	Socialisation	The organisation

Contexts of knowledge

Context	
Domains	More or less well-defined 'parts of the world' to which a particular body of knowledge applies
Situations	Assemblies of components, domains, people and other elements (or 'human and non-human carriers of knowledge' [Hardstone1998]) present at any particular instant of expert activity (or 'knowledge mobilisation' [12])
Milieus	The immediate environments in which expertise is exercised; comprising sets of situations occurring regularly at particular locations, e.g. laboratories, operating theatres, offices, etc.

Each individual, as a carrier of knowledge, can know the same thing or concept simultaneously in different ways (as different cognitive components). Thus practitioners may have formal knowledge about an aspect of their domain, but this will be internalized and amplified through experience, practice and local conditions to create informal, tacit and contingent knowledge. Hence the relative importance of each cognitive component to carriers changes over time and space. Dealing with tacit knowledge is perhaps not such a problem (for system designers, for instance) after all, as other non-formal knowledge components may be at least partially articulable.

By combining taxonomies of knowledge [5, 11, 20] that relate to the cognitive content of knowledge [5, 11] and the substantive content of domain knowledge in practice [11, 20], extended to include social and organisational knowledge [11], the knowledge related to a particular domain or task within that domain may be conceptualised as distributed across a grid, each square of which tells us something about the social nature of that knowledge. An example is shown in Table 3.

Table 3. The substantive and cognitive content of knowledge: Grid for analysis (of a particular activity, domain, situation or milieu – for example, bed management).

Substantive **Cognitive**	Fundamental concepts (what it is; how it works; 'normal')	Criteria and specifications (quantitative goals)	Theoretical tools (maths methods; intellectual concepts)	Quantitative data (descriptive and prescriptive)	Practical considerations (incl. judgement)	Design instrumentalities (knowledge about procedures)
Formal knowledge						
Informal knowledge						
Contingent knowledge						
Tacit knowledge						
Instrumentalities						
Meta-knowledge						

To operationalise these concepts in a domain context, we can conduct grid and then gap analyses based on taxonomies of knowledge [5, 11, 20], identifying the knowledge being mobilised and its distribution, and charting the networks of people and objects involved in particular tasks or activities. The outputs constitute useful analytical tools, particularly when translated into diagrammatic representations. We can compare the old and new systems, identifying differences and problem areas from a knowledge perspective. This information can be fed back iteratively to designers and users, using clear and accessible representations where appropriate. These serve as communication artefacts or 'translators' [11, 12, 20] to support communication between domains.

Domain knowledge is varied in content and unevenly distributed within the socio-cognitive structures [5] of technological systems, with significant overlaps between carriers. This distribution is shaped both by structural and by more contingent social factors, which can be described and analysed in sociological terms. It can also be mapped onto the squares of the grid described above, and a gap analysis conducted to discover whether unpopulated squares are either irrelevant or problematic, and which squares contain knowledge crucial to specific tasks or activities. By looking at how knowledge is put into practice in context within an organisation, it is possible to discover both how knowledge is distributed, and how it is mobilised for specific activities, including those supported by computer systems. The mobilisation of knowledge is almost always collective, occurring through the formation of temporary networks of human and non-human carriers. Over time, with repetition, some of these networks and mobilisation processes become institutionalised in communities and routines, independent of the specific individuals involved. These networks and groupings can be described and represented diagrammatically.

To operationalise these concepts in a systems design and configuration context, we can chart the diverse networks and communities of carriers assembled for particular tasks or activities, whether routine or ad hoc. We can combine this with use of the grid to identify the kinds of knowledge that are being mobilised and how they are distributed. The outputs (grids and network charts) should provide a useful analytical tool, particularly when combined with, or translated into diagrammatic representations. By conducting such an analysis of both the old and the new systems, we can compare the socio-technical system that is being replaced with the new system; identifying where the differences lie, and where problem areas might be, from a knowledge perspective. This information can be fed back iteratively to both designers and users, using clear and accessible diagrammatic representations where appropriate, as described below, to assist communication and discussion.

Studies touching on the mobilisation of knowledge by carriers from more than one domain [11, 12, 20], such as between system designers and domain users, suggest that some of the people and also the artefacts involved need to be able to operate in the networks of both domains, acting as ‘translators’ (especially in the light of domain-specific languages) for problem-solving to occur. The use of diagrammatic representations in the proposed empirical context may provide one means of translating between domains.

We propose to combine the above with previous research addressing the issue of communication concerning the design, assessment and deployment of complex, highly dependable computer-based systems, where that communication must take place across technical and non-technical boundaries. In that context, as here, knowledge concerning the (evolving) design, and the impact of changes to it, is distributed across a broad range of stakeholders representing multiple technical and non-technical disciplines, who hold diverse needs and goals. Our previous research has extensively studied the role of differing forms of representation, particularly diagrammatic, in facilitating the communication of knowledge in this context.

3 Representing Knowledge

The use of diagrammatic representations is common throughout engineering and design practice [8]. Previous research has compared diagrammatic and textual forms of representations from both semantic and cognitive perspectives [10]. This work will inform our design of diagrammatic languages and notations to capture the implications of our social analyses in accessible forms, facilitating the communication of knowledge during the design and deployment of complex, highly dependable computer-based systems across technical and non-technical domain boundaries. Thus we aim to not only study the potential impact of the proposed technological system, but also to make this impact visible and accessible to a broad range of stakeholders through the use of appropriately designed representations.

3.1 Designing effective diagrammatic representations of knowledge

Diagrams are popular, as many people find them more readily "accessible" than other forms of representation. Diagrams are also effective at presenting "the big picture"; that is, diagrams can typically contain far more visible structure than any text-based representation and this structure can be used to reflect the structure of whatever it is that the diagram represents. Diagrams are thus particularly popular and effective in design, where they are typically most effective at presenting high level overviews of entire systems, in which the relationships and interactions between components is highly visible, and thus more readily accessible.

An illustrative example of the significance of a well chosen representation in facilitating communication across technical boundaries for highly dependable systems is the HAZOPS (HAZard and OPerability Studies) hazard analysis technique [1]. HAZOPS is a technique that originated in the chemical industries which involves engineers and experts from a broad range of technical 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, schematics of the physical plant layout (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. The diagram thus plays the role of a *communication artifact*, an entity which guides and supports communication concerning the system under analysis.

An effective diagram is typically taken to be one that is "well matched" to what it represents. This is to say, that the logical and spatio-visual properties of structures inherent to the diagram are chosen so as to have some very direct correspondence with the structures that they represent in the semantic domain; and in particular that they are chosen so as to support desired reasoning tasks by making certain inferences *immediate* and *obvious*. A more detailed exploration of this issue, including a formalisation of the concept of well matched, is in [10].

In this section we present guidelines for both the design of effective diagrammatic languages, and the design of specific diagrams within such languages. These guidelines draw upon results from visual language theory, cognitive science, empirical psychology and graphic design. Integrating results from such diverse fields is a non-trivial task, which is here approached through a decomposition of the study of issues of effectiveness in diagrammatic languages according to analogous understandings of (written and spoken) natural languages. We present on overview of this study next.

3.2 Exploring diagrammatic "matching"

The study of natural languages is typically separated into the following categories: phonetics and phonology; morphology; syntax; semantics; pragmatics; and discourse. With the obvious exception of the first, the study of analogous categories in diagrammatic languages is at the same time both highly revealing of differences and similarities between the two forms of representation; and also provides a structure in which to explore the alternative means by which a diagram may capture meaning. Separating the study of diagrammatic languages into these categories permits us to firstly lay out the various means by which the structure inherent to diagrammatic morphologies and syntax may directly capture structure in the semantic domain; and secondly to consider how further pragmatic usage may convey meaning in diagrams. Such a study is undertaken in [10], which extends earlier work of [9] in decomposing the variety of issues pertaining to effectiveness in diagrams. This section presents an overview of this exploration, focusing on the alignment of syntactic features of diagrams to their semantics.

Morphology concerns the shape of symbols. The shape of a particular alphabetic character cannot convey much variation in meaning; an 'a' is an 'a' regardless of its font or whether or not it is bold or italicised. By contrast, the basic vocabulary elements in some diagrammatic language may include shapes such as circles, ellipses, squares, arcs and arrows, all of differing sizes and colours. These objects often fall naturally into a hierarchy which can constrain the syntax and, furthermore, inform the semantics of the system. This hierarchy may be directly exploited by the semantics of symbols so as to reflect the depicted domain.

A number of studies such as [3, 17] have attempted to categorise diagrammatic morphology, Horn [13] reviews these and proposes a unified categorisation

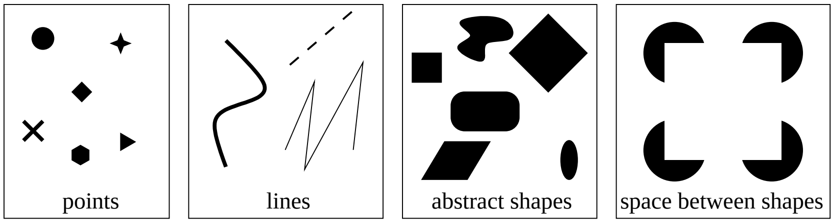

Fig. 1. Morphology of Shapes (Horn'98).

(for generic representations) whose most general categories are: words; shapes; and images. Here we focus on shapes, which Horn subdivides into: points; lines; abstract shapes; and "white space" between shapes – although we do not consider this latter here.

The category of abstract shapes, and potentially that of shaped points, may be further subdivided. For example, regular shapes may be divided into "smooth" and "angled" as determined by their corners. Such sub-categories may be further divided, leading to a type-hierarchy of shapes which may be directly exploited by the semantics of symbols so as to reflect the depicted domain. For example, consider a map on which cities are represented as (shaped) points. A categorisation of points divided into smoothed and angled could be exploited by a corresponding categorisation in th semantic domain with, say, smoothed points (circles, ellipses, etc) representing capital cities and angled points (triangles, squares, etc) representing non-capital cities. The division of smoothed and angled points into further sub-categories could similarly correspond to further sub-categorisations of capital and non-capital cities. Note however that there is no unique *canonical* hierarchy of shapes.

In addition to a morphological partial typing, symbols may be further categorised through graphical properties such as size, colour, texture, shading and orientation. For example, the meaning of symbols represented by circles may be refined by distinguishing between large and small, and different coloured circles. Thus, again, part of the structure in the semantic domain is directly captured by morphological or syntactic features. [1] The properties of graphical symbols we consider here – again modifying those suggested in [13] – are: value (e.g. grey-scale shading); orientation; texture (e.g. patterns); colour; and size. These are applied to points, lines and shapes as in Table 4.

In addition to exploiting the structure of the morphology of diagrammatic symbols, we may also exploit the structure and properties inherent to diagrammatic syntactic relations in ensuring that a diagram is well matched to its meaning. For example, the use of *inclusion* or *overlap* to represent semantic relationships which share logical properties with these syntactic relations. A promising exploration of the properties of various syntactic diagrammatic relations (primar-

---

[1] Note that textual tokens may also display such properties in a slightly more limited sense, such as font, italics, etc.

Table 4. Properties of primitives (2)

	Value	**Orientation**	**Texture**	**Colour**	**Size**
Point		min		X	
Line	lim	X		X	X
Shape	X	X	X	X	X

ily of relations between pairs of diagrammatic objects) is given by von Klopp Lemon and von Klopp Lemon [21], who define the logical characteristics of 12 properties and examine their presence or absence in around 65 syntactic diagrammatic relations.

Finally, in linguistic theories of human communication, developed initially for written text or spoken dialogues, theories of pragmatics seek to explain how conventions and patterns of language use carry information over and above the literal truth value of sentences. Pragmatics, thus, helps to bridge the gap between truth conditions and "real" meaning - that is, between what is *said* and what is *meant*. This concept applies equally well to diagrams. Indeed, there is a recent history of work which draws parallels between pragmatic phenomena which occur in natural language, and for which there are established theories, and phenomena occurring in visual languages - see [15] for a review of these.

3.3 Guidelines for diagram language design

Our guidelines for diagrammatic language design are as follows:

1. identify the fundamental semantic concepts and any structuring which exists over these. Match this to the morphological structure of graphical primitives;
2. identify features and properties of these semantic concepts and match to properties of the chosen symbols and graphical syntactic features;
3. identify properties of semantic relationships between objects and match these to syntactic relations.

However, this matching must be in the context of consideration of the tasks which the potential diagrams are intended to support. These tasks should indicate the key features, and the syntax should be chosen so as to achieve maximum salience of these. This desire will also inform decisions when there is a choice of equivalent syntactic matches for some desired semantic feature.

Note that as certain graphical properties and syntactic relations may interfere, often a balance or trade-off is required when selecting the most appropriate syntactic match for some semantic aspect. Experience in graphic design (e.g [18, 19]) suggests a rule of thumb that *task* concerns outweigh *semantic* concerns; that is – where a trade-off is required, the preference should be whichever option supports greater salience of task-specific features.

Typically, for any non-trivial semantic domain and intended tasks, not all information may be captured directly through diagram syntax. Consequently the use of *labelling languages* for labels which may potentially contain significant semantic information is necessary for most practical diagrammatic languages. However, in an effort to increase the expressiveness, the unprincipled use of sophisticated labelling languages can perturb the directness of a diagrammatic language. Examples of languages which are diagrammatic at core, but have had their expressiveness enhanced through sophisticated labelling languages until any benefit to readers interpretation of the "diagrammatic aspects" is negated, are legion. This is a substantive and open issue which is beyond the scope of this paper, and so we merely issue the warning: treat labels with care.

Finally, the construction of any specific diagram must also ensure that any non-semantic aspects are normalised as far as possible, as random or careless use of colour or layout, for example, can lead to unwanted mis- or over-interpretation by the reader.

3.4 An example of "well-matched" diagrams

One practical application of the guidelines proposed above appears in a study by Oberlander *et al* [16] of differing cognitive styles in users of the computer-based logic teaching tool Hyperproof [2]. A language was devised for [16] which provided the reader with a salient and accessible representation of the significant differences in the use of Hyperproof by the two groups, named "DetHi" and "DetLo".

Examination of this semantic domain suggested that a simple node-and-link representation, where nodes represented Hyperproof rules (user commands), and directed links represented the "next rule used" relationship, captured the key concepts. The features seen as most necessary for presentation to the reader were the frequencies both of rule use and of transitions between specific pairs of rules. The preferred matching of these features to properties of boxes and arrows, as indicated by Table 4, was the use of *size* to represent frequency in each case. Thus the relative size of nodes directly corresponded to the relative frequency of rule use. Following the above guidelines, lines were restricted to being one of five discrete sizes, with increasing size indicating increasing frequencies. Thus each specific line width represented a range of frequencies relative to the issuing node, with frequencies of 10% and lower not being represented. Absolute transition frequencies are therefore represented by accompanying textual labels. The resulting diagrams are repeated here in Figures 2 and 3.

The final consideration for the construction of these two specific diagrams in the devised language concerned the use of layout. The tasks for which the diagrams were to be put were of two kinds: the identification of patterns in a single diagram; and the identification of characteristic differences between two diagrams. Layout had a mild impact on the former task, suggesting that as far as possible the layout should place connected nodes in spatial proximity. Layout had a greater impact on tasks of the latter kind, suggesting that to facilitate comparisons firstly the layout of nodes in the two diagrams should be as similar as possible; and secondly that where size (area) of a node varied between the two diagrams, this variance should take place along a single dimension wherever

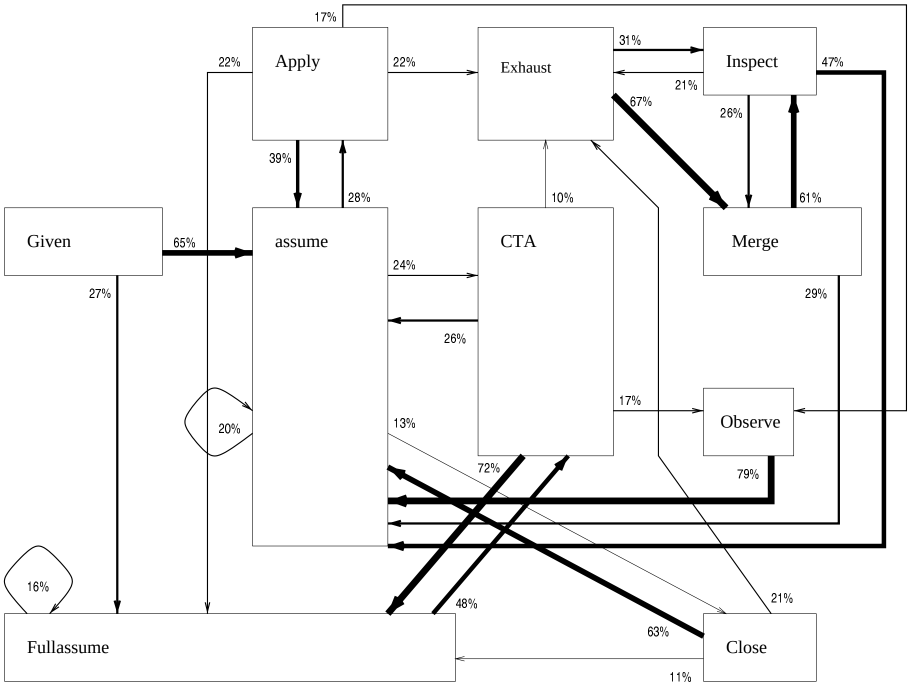

Fig. 2. Transition network for DetHi behaviour on indeterminate questions.

possible (in accordance with the relative perceptual salience of comparison along identical uni-dimensional scale versus area, as indicated in empirical psychological studies such as [4]). One final point of note is that Hyperproof's `Close` rule was never used by DetLo subjects. Following the guideline that *task* concerns outweigh *semantic* concerns, the pragmatic decision was made that the `Close` node should be represented in Figure 3 (rather than being of zero size). However, to indicate that this node categorically differed from all other nodes in that diagram, its bounding line was represented with a lesser value (i.e. a dashed line).

The effectiveness of this diagrammatic language for the required tasks should be readily apparent to the reader. Note, for example: the characteristic differences between the use of the `Observe` rule by DetHi and DetLo subjects; patterns of rule use such as `Merge-assume` by DetHi subjects which are completely absent in DetLo subjects; and the generally more "structured" use of rule-pairs by DetHi subjects – indicated by the greater number of thick lines, and fewer lines overall, in Figure 3.

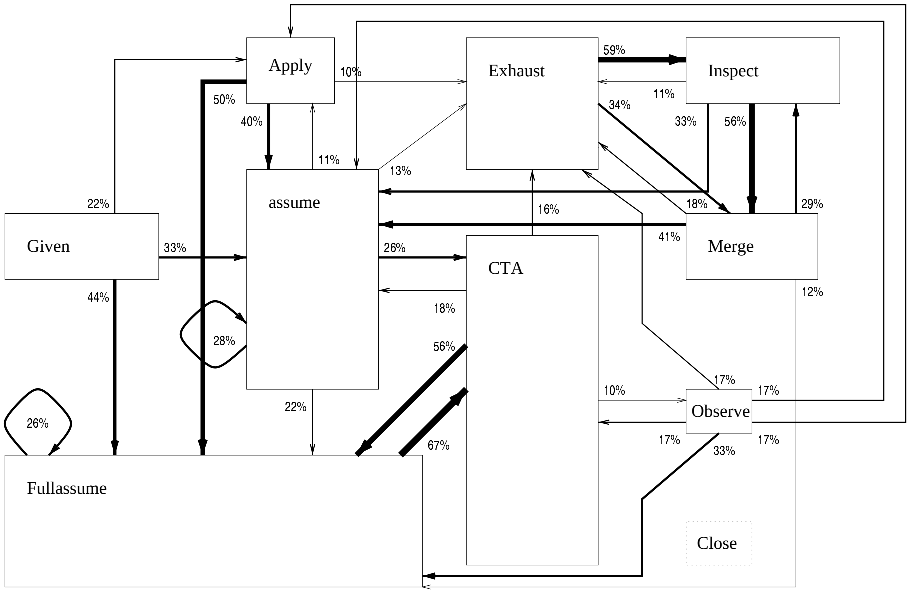

Fig. 3. Transition network for DetLo behaviour on indeterminate questions. Note that `Close` is not visited at all

4 Summary

Our initial application and evaluation of this work is in the domain of Healthcare Informatics. We will be working with a large NHS hospital, which is in the process of designing and implementing a new computer-based Health Information System (HIS). As various modules of the HIS are implemented, we will compare pre- and post-HIS working practices. Using the methods outlined above, we will provide feedback between users and designers throughout the design and implementation process.

The organizational structure of a hospital is typically one of great complexity and the needs and knowledge of system users are significantly diverse. In combination with the expectation that the proposed system will be subject to substantial local configuration for different medical and administrative departments, it is clear that the integration of this system into the existing hospital environment offers a fruitful opportunity for us to evaluate the efficacy of both our representations and our overall analytical approach to this task.

References

1. Chemical Industries Association. *A Guide to Hazard and Operability Studies.* 1992.
2. J Barwise and J Etchemendy. *Hyperproof.* CSLI Publications, 1994.

3. J Bertin. *Semiology of graphics: Diagrams, networks and maps.* University of Wisconsin Press, Madison, WI, 1983.
4. W S Cleveland. *The elements of graphing data.* Wadsworth, Pacific Grove, CA, 1985.
5. J Fleck. Innofusion or diffusation? the nature of technological development in robotics. PICT Working Paper Series 4, University of Edinburgh, 1988.
6. J Fleck. Configuration: Crystallising contingency. *International Journal on Human Factors in Manufacturing*, 1992.
7. J Fleck. Expertise: Knowledge, power and tradeability. In Williams et al, editor, *Exploring expertise: Issues and perspectives.* Macmillan, 1998.
8. C Gurr. Knowledge engineering in the communication of information for safety critical systems. *The Knowledge Engineering Review*, 12(3):249–270, 1997.
9. C Gurr, J Lee, and K Stenning. Theories of diagrammatic reasoning: distinguishing component problems. *Mind and Machines*, 8(4):533–557, December 1998.
10. C A Gurr. Effective diagrammatic communication: Syntactic, semantic and pragmatic issues. *Journal of Visual Languages and Computing*, 10(4):317–342, August 1999.
11. G Hardstone. *Robbie Burns' moustache: print knowledge and practice.* PhD thesis, University of Edinburgh, 1996.
12. G Hardstone. You'll figure it out between you: Problem-solving with the web-8. In Williams et al, editor, *Exploring expertise: Issues and perspectives.* Macmillan, 1998.
13. R E Horn. *Visual Language: Global Communication for the 21st Century.* MacroVU Press, Bainbridge Island, WA, 1998.
14. D A MacKenzie and J Wajcman, editors. *The social shaping of technology.* Open University Press, Buckingham, 2nd edition, 1999.
15. J Oberlander. Grice for graphics: pragmatic implicature in network diagrams. *Information design journal*, 8(2):163–179, 1996.
16. J Oberlander, P Monaghan, R Cox, K Stenning, and R Tobin. Unnatural language processing: An empirical study of multimodal proof styles. *Journal of Logic Language and Information*, 8:363–384, 1999.
17. F Saint-Martin. *Semiotics of visual language.* Indiana University Press, Bloomington, IN, 1987.
18. E R Tufte. *The Visual Display of Quantitative Information.* Graphics Press, Cheshire CT, 1983.
19. E R Tufte. *Envisioning Information.* Graphics Press, Cheshire, CT, 1990.
20. W G Vincenti. *What engineers know and how they know it: Analytical studies from aeronautical history.* John Hopkins University Press, Baltimore, MD, 1990.
21. A von Klopp Lemon and O von Klopp Lemon. Constraint matching for diagram design: Qualitative visual languages. In *Diagrams 2000: Theory and Application of Diagrams*, LNAI 1889, pages 74–88, Berlin, 2000. Springer.