Argumentation in Decision Support for Medical Care Planning for Patients and Clinicians

David W. Glasspool, John Fox, Ayelet Oettinger, James Smith-Spark
Advanced Computation Laboratory
Cancer Research UK
PO Box 123, London WC2A 3PX, United Kingdom.
{david.glasspool, john.fox, ayelet.oettinger}@cancer.org.uk, jss@acl.icnet.uk

Abstract

Developing a care plan for a patient requires an understanding of interactions and dependencies between procedures, and of their possible outcomes for an individual patient, and it requires the planner to keep track of this information as the proposed plan evolves. This is difficult even for experienced clinicians, but increasingly patients are expected (and expect) to participate. We describe an argumentation-based planning support system designed to ameliorate the cognitive load imposed by the planning and communication elements of such tasks. An initial evaluation study in the field of genetic counseling produced promising results. The approach may provide a general aid for clinicians and patients in visualizing, customizing, evaluating and communicating about care plans.

1. Introduction

The problem of communicating and working with risk and uncertainty is a serious one for many fields. It is a particular issue in medicine because of the speed at which medical knowledge is accumulating, the level of uncertainty inherent in the field, and the increasing desire to include patients (who of course have no training either in medicine or in decision-making under uncertainty) in the planning of their own care. Much work has been done developing decision support systems for single, isolated medical decisions (for example, what drug to prescribe, or whether to refer a patient to a specialist), and argumentation logic approaches have been shown to be effective in communicating complex uncertain information to both patients and clinicians in such cases (Fox, Glasspool and Bury, 2001). However in medicine most actions are not undertaken in isolation but form part of a care plan for a patient, where they are likely to depend on the results of earlier decisions and interact with later events and decisions in complex ways.

For example, consider a woman diagnosed as carrying a mutation to the gene BRCA1 which confers a high risk that she will develop breast or ovarian cancer during her lifetime. Following a positive gene test (now widely available for women with a suspicious family history) the woman will typically be seen by a genetic counsellor who explains her situation and what options are available for mitigating the risk. These may include prophylaxis with drugs such as Tamoxifen, screening to detect tumours early in development, and pre-emptive surgery to remove the ovaries (oophorectomy) or breasts (mastectomy).

There is no “correct” plan of action here, the care plan arrived at by the counsellor and patient will reflect the individual needs and plans of the patient. For example, she may be planning to have children, and will need to avoid oophorectomy and drugs like Tamoxifen until there is no further chance of pregnancy. People vary widely in their willingness to consider mutilating surgery like mastectomy. The timing of oophorectomy and some drugs relative to each other and to menopause can affect their action. Each decision involved in forming the plan is potentially influenced by previous decisions, and in turn influences later decisions. The available options, and the information needed to evaluate them, change continually as the plan is modified. Experimental evidence (Klein, 1998; Smith-Spark et al. 2005) and consideration of the likely cognitive loads imposed by plan manipulation (Glasspool et al. 2003) indicate that working with medical care plans is a demanding task even for experts in the field, let alone the patients whose care is at stake.

In this paper we describe a software application, REACT, based on argumentation logic, which provides support for clinicians and patients engaged in the type of medical planning described above. In the next section we describe the REACT user interface and show how it is motivated by the cognitive processes involved in planning. In section 3 we discuss the role of argumentation in providing this interface, and in section 4 we report evaluation results showing that clinicians find the system effective in joint care planning with patients. Finally we discuss some shortcomings of the present approach and plans for future development.

2. The REACT system

Relatively little work has been done on understanding the cognitive processes involved in planning (see Morris and Ward, 2005). It is known that planning makes large demands for cognitive resources and involves a number of cognitive processes (e.g. Cohen, 1989). The cognitive demands are related to the number of variables and interactions within a plan (Klein 1998, p52), and the level of detail in which a planner considers the future consequences of actions influences their plans (Huys, Evers-Kiebooms and d'Ydwalle, 1992; Hirt & Sherman, 1985). Finally, there is evidence that computer displays can aid performance in cognitively demanding tasks by making constraints and possibilities inherent in the problem domain implicit in the display (e.g. Stenning and Oberlander, 1995, Zhang and Norman, 1994). Our own work has shown that the load on short-term working memory is a significant factor limiting performance in plan manipulation tasks, and that appropriate computer interface design can remove much of the burden on working memory resources (Smith-Spark et al. 2005). Based on these results we have proposed (Glasspool et al. 2003) that the major sources of difficulty for an unaided person manipulating a complex plan involve:

1. Maintaining an evolving plan in memory as it develops.
2. Identifying which options for action are available at each step in the plan.
3. Deciding which of these options should actually be taken.
4. Keeping track of constraints on, and dependencies between, planned actions.
5. Keeping track of the effect of the plan as a whole with respect to its goals.

We have developed a computer-based planning support tool, REACT (Risks, Events, Actions and their Consequences over Time), which provides four distinct types of decision support designed to allow the cognitive loads listed above to be "offloaded" (Scaife & Rogers, 1996) to a graphical interface (figure 1):

- An interactive chart showing past and expected events and the actions that are currently planned laid out over a time line. This chart allows the user to offload *memory for the current plan configuration* and *identification of options for action.*
- Visual feedback of conflicts between planned actions and constraints on actions is given immediately while actions are being manipulated on the planning chart. This allows the user to externalize (i.e. rely on the software for) *calculation of temporal constraints and dependencies* as planning progresses.
- Feedback of the predicted effect of the developing plan on quantitative outcome measures such as risk is given immediately during planning. In the present version of the software this is provided by a graph of risk (or other outcome) against time which is continually updated while the user interacts with the software, though other types of display (e.g. showing frequencies rather than probabilities, or using different visual approaches) are possible. This allows users to externalize *prediction of the consequences* of actions and plans.

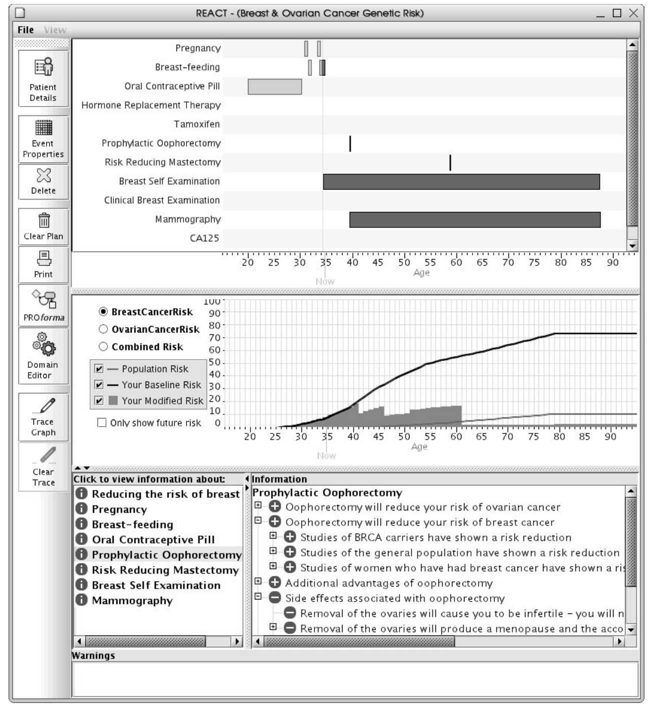

Figure 1: *The REACT user interface. At the top of the screen a plan is being developed for risk mitigation in the domain of genetic predisposition to breast cancer by manipulating actions on a planning chart. The central graph indicates the estimated risk of death due to breast cancer (assuming the current plan were carried out) while arguments for and against prophylactic oophorectomy for this patient are reviewed in the argumentation area in the lower part of the screen.*

- Logical arguments for and against each proposed action are automatically displayed in real time. This allows the planner to review reasoning for specific decisions.

Overall REACT provides a summary of the individual clinical circumstances, and a "what if" view of the proposed care plan. As the user adds, removes or moves events on the planning chart at the top of the screen, the decision support system provides real-time feedback in the rest of the display showing the expected consequences of the action. This allows the user to easily explore the "space" of options available within the plan with immediate feedback of the various interactions and consequences. Information directly relevant to each option is summarized in an accessible form based on arguments for and against the action.

3. Argumentation in REACT

We have found an argumentation logic approach to be effective in communicating information about uncertainty to clinicians and patients (Fox, Glasspool and Bury, 2001). Argumentation appears to be particularly appropriate in this application, where potentially complex relationships and interactions must be represented. We therefore set out to represent and process all information in the REACT application in the form of logical arguments. This information includes verbal arguments for and against planned treatment options, conflicts and dependencies between events, and numeric information (for example, graphs of risk, as shown in figure 1). Our general approach to argumentation (Fox and Das, 2000) treats quantitative information and processing (such as traditional Bayesian probability values and calculations) as special cases of the more general logical framework (Fox, Glasspool and Bury, 2001), allowing numerical probability calculations to be represented within the same general framework as qualitative arguments where appropriate quantitative information is available.

Figure 2 shows the computational architecture of the system. A *domain knowledge base* specifies all knowledge relating to a particular clinical application, comprising a set of arguments for and against various claims. Arguments may relate to reasons for or against planning particular actions, to the effect of actions on outcome measures (for example overall risk or cost of the plan), and to ordering constraints between actions or events in a plan. For example arguments may be made that particular actions or events must or must not occur before, after or simultaneously with other actions or events.

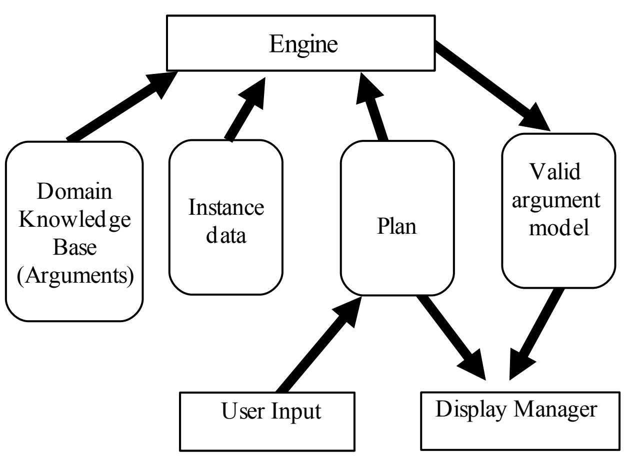

Figure 2: REACT computational architecture. Ellipses represent data, rectangles represent computational process.

The concept of argumentation as an alternative way of modeling everyday reasoning to more technical concepts like mathematical logic and probability is usually attributed to Toulmin (1957) but has recently been widely taken up in computer science and artificial intelligence (e.g. for modeling legal and medical reasoning) and in psychology under the heading "reason based decision making" (Curley and Benson, 1994). An important feature of Toulmin's model, which we have adopted, is the explicit treatment of the direct logical justification for an argument (which he calls the "warrant") and the background knowledge ("backing") which provides the deeper rationale. Arguments in REACT have the following form:

Condition --> (Type, Text, Arithmetic function, Backing)

Where "Condition" is a logical condition that is tested against the current state of the plan to determine if the argument is valid. It corresponds to Toulmin's concept of the *data* upon which an argument is based.

Taken together, the "Text" and "Arithmetic function" fields constitute the claim of the argument in Toulmin's terms. The "Text" field contains text to be displayed to the user if the argument is valid (i.e. if the condition is true), while the "Arithmetic function" field can optionally provide a function to be applied to a specific numerical value (such as a risk graph) if the argument is valid.

The "Type" field indicates whether the argument counts in favour of the claim or against it (types "Pro" and "Con"). It is also possible for an argument in REACT to provide useful information that is neither obviously for or against the claim, in which case an additional argument type - "Information" - is available.

In the present version of REACT each argument is evaluated independently, and the emphasis is on providing all relevant information to a user making a decision, rather than advising on what decision should be taken. The "Type" information is therefore used to organize the text display of the argument and determine what visualization should be included in the graphical display if the argument is valid, rather than to control an inference process. It corresponds to what is sometimes referred to as the "sign" of the argument (Fox and Das, 2000) (or roughly to Toulmin's *qualifier*, but see the discussion of this issue in section 5).

Finally an arbitrary number of "Backing" facts may be associated with an argument, each of which comprises text to be displayed to the user and, optionally, a URL to be opened in a web browser if the user wishes. These correspond to the authority upon which the argument is made, and typically comprise references to studies in the medical literature.

The entire structure, associating a logical condition with a claim, effectively comprises the logical warrant for the claim in Toulmin's terms. For example, a typical argument in the breast and ovarian cancer genetic counseling domain is as follows:

Condition: "Pregnancy" appears in the plan.

Type : "Pro"

Text : "Pregnancy reduces risk of ovarian cancer".

This will cause the text "Pregnancy reduces risk of ovarian cancer", labeled as an argument in favor of pregnancy, to appear if pregnancy appears anywhere in the plan.

The user interacts with the system by modifying the displayed plan. This updates the plan data, which triggers the engine to recalculate the set of valid arguments. The display manager then updates the display to reflect the new valid argument set. All of this happens in real time as the user moves, inserts or deletes actions in the plan.

4. Evaluation

A preliminary evaluation has been carried out of REACT in use in a simulated genetic counseling setting, using a scenario very similar to that described in the introduction. The participants were eight cancer genetic counselors at Guy's Hospital, London. Each was given brief initial training before two testing sessions, in each of which they counseled a woman carrying a risk-increasing gene mutation, played by a professional actor. In one session the REACT software was used. Participants' attitudes towards REACT and its use during counseling were assessed by questionnaire and semi-structured interview.

The participants saw a number of benefits to REACT. They found that the dynamic visual display put across information clearly and concisely, and that logical arguments provided "bullet-points" for discussion. They appreciated accurate, detailed, and up-to-date information tailored to the specific patient. Access to information was felt to be empowering, and two counselors commented that the software helped them to improve the structure of their consultations. They found the software easy to use and to learn. Although most participants were initially somewhat skeptical about using any form of computer aid in counseling sessions, after the trial seven of the eight participants were very positive about using REACT clinically, and all felt it to be worthwhile.

A number of concerns were raised. Some related to problems participants expected patients to have with the information display, for example the interpretation of graphs. Participants were concerned about possible changes in the dynamics and content of the consultation, and were unsure how much information should be given to the patient.

Encouragingly, the participants themselves offered solutions to the problems that they raised. They identified the need to be selective in the use of REACT, both in terms of which types of patients would benefit and when within the counseling session the computer should be used. Participants also felt that they would need to acquire some new counseling skills in order to use the software effectively. However overall the benefits of REACT were felt to make this investment worthwhile.

5. Discussion

The present REACT interface allows the user to work only on a single plan, exploring the consequences of variations in the plan. However, once the number of possible variations becomes large it would clearly be valuable to be able to maintain and compare alternative versions of a plan within the user interface. We propose to add functionality that would allow a plan to be "cloned", an identical copy being placed on a separate "tab" on the user interface. Switching between the tabs the user may then explore the effect of multiple changes in the new copy compared to the original. In principle any number of variants could be created and instantly switched between. In the present version of the interface a related but much more limited function is available: The user may create a permanent "trace" of the current outcome measure graph, which remains visible and static as the plan is changed so that differences in the graph compared with the point at which the trace was made are evident.

In the current version of the software the argumentation and other displays are only used to *inform* planning and decision-making, not to *recommend* specific decision options, but the model can be extended to support the assessment of the options. For example we can generate a measure of the relative "force" of argument for the competing options by comparing the number of pros and cons for each option in the simplest case.

More generally, arguments can be weighted in various ways (Fox, Glasspool and Bury, 2001). While the "Type" field of a REACT argument corresponds very roughly to Toulmin's notion of a qualifier (in that it indicates at least the sign of an argument, if not its strength), the more general form of argumentation we have proposed (Fox and Das, 2000; Fox, Glasspool and Bury, 2001) includes the notion of a qualifier drawn from a specified *dictionary* of terms, with a corresponding aggregation function that specifies how a group of arguments for or against a single claim should be combined. The minimal dictionary {Pro, Con}, with an aggregation function that counts the number of "Pro" arguments and subtracts the number of "Cons", corresponds to the simple case mentioned above and has been successfully used in a number of medical decision support applications (Fox, Glasspool and Bury, 2001). Another possible dictionary is the set of real numbers from 0.0 to 1.0, which would allow more fine-grained weighting of arguments. In particular qualifiers could then be specified as numerical probability values, in cases where these are known, and with an appropriate aggregation function based on the theorems of classical probability this would allow the scheme to reduce to standard probability theory.

A more serious limitation of the argument model used in the current version of REACT is that all knowledge in the system is stored as pre-compiled arguments. This approach has the advantage of simplicity (and has proved effective in the clinical applications we have developed so far) but it presupposes that the knowledge author has anticipated all the applicable arguments. In particular more complex argument structures (such as undercutting, rebutting or supporting arguments), which one might wish the system to construct automatically, are difficult to implement (and not implemented in current knowledge bases). Moreover it is not possible to *construct* arguments "on the fly" from knowledge. To illustrate, suppose that a user is investigating the option of oophorectomy. It is straightforward to define, in the REACT knowledge base, an argument against oophorectomy in a pre-menopausal woman on the grounds that it causes infertility. However this presupposes that infertility is an undesirable outcome. If the woman does not wish to have further children she may be indifferent to this outcome (or may even see it as an *advantage* of oophorectomy). The fixed, pre-compiled nature of the current argument scheme also means that updating the knowledge base to reflect new clinical knowledge (new, more solid data on side effects of a drug, for example) can be a non-trivial task if many arguments in the knowledge base use that clinical knowledge in different ways.

A more flexible approach would be to store medical knowledge, rather than arguments, in the form of declarative facts (of the form "oophorectomy in a pre-menopausal woman causes irreversible infertility"), along with a set of standard argument schemas (for example, "if action A is expected to have effect E, on the basis of knowledge source S, and E is undesirable, then E constitutes an argument against A, warranted by S"). With appropriate knowledge about the desirability or undesirability of various types of outcome (perhaps a set of default values which could be overridden by personal preference) the system could then construct more appropriate arguments dynamically. Updating the knowledge base to reflect new medical knowledge would also be a simpler and safer proposition. (An interesting side issue is that once the argumentation model allows arguments to be constructed dynamically in this way, the structure and granularity of the knowledge base will become interesting. There would be benefits to organising the knowledge as a hierarchical ontology, so that rather general argument schemas could be written which could instantiate against a range of much more fine-grained details in the knowledge base as required.) We are currently participating in a European project ("ASPIC") which aims to develop a general model for argumentation services, and we plan to use work on argument representation formats and argumentation engines under that project to provide these features in REACT.

We see the REACT system as potentially useful over a wider range of applications than genetic counseling, and with a variety of different modes of use. Generally, we believe that the approach will be an effective way to enable people to understand and manipulate clinical plans in the increasing number of situations where this is important. Counseling emphasizes communication between counselor and patient during planning. In other situations a clinician might use the software alone (in treatment planning for a single patient, or authoring a standard plan for treating a condition), or it might be used jointly by a group of clinicians discussing treatment options for a patient. Alternatively the software might be used by the patient alone (in situations where the need for support and interpretation by a qualified clinician was less strong than in genetic counseling, of course). One priority for future work is to assess the use patterns and requirements for these types of application.

Acknowledgments

This research was supported by award L328253015 from the UK Economic and Social Research Council and Engineering and Physical Sciences Research Council under the PACCIT programme. Software implementation and definition of medical knowledge bases was carried out in part with funding from Cancer Research UK. We are grateful to Sanjay Modgil, Jon Bury and Peter McBurney for discussion of the argumentation model, and to Fred Kavalier, Tracy Bussoli and Cindy Zaitsoff and their colleagues in the Department of Clinical Genetics, Guy's Hospital, London for their help with the evaluation study.

References

Cohen, G. 1989. *Memory in the real world.* Hove, UK: Psychology Press.

Curley, S. P. and Benson, P. G. 1994. Applying a cognitive perspective to probability construction. In G Wright & P Ayton (eds.), Subjective Probability. Chichester, England: John Wiley & Sons. pp. 185-209.

Fox, J. and Das, S. 2000. *Safe and Sound: Artificial intelligence in hazardous applications.* Cambridge, Mass.: MIT press.

Fox, J., Glasspool, D. W. and Bury, J. 2001.

Quantitative and Qualitative Approaches to Reasoning under Uncertainty in Medical Decision Making In: S. Quaglini, P. Barahona, S Andreasson (Eds.): *Proceedings of the 8th Conference on Artificial Intelligence in Medicine in Europe (AIME 2001)*. Springer-Verlag, Berlin. pp. 272-282.

Glasspool, D. W., Fox, J., Castillo, F. C. and Monaghan, V. E. L. 2003. Interactive Decision Support for Medical Planning. In M. Dojat, E. Keravnou and P. Barahona (Eds.): *Proceedings of the 9th Conference on Artificial Intelligence in Medicine in Europe, AIME 2003.* Lecture Notes in Artificial Intelligence (LNAI 2780), Berlin: Springer-Verlag. pp. 335-339.

Hirt, E. R., & Sherman, S. J. 1985. The role of prior knowledge in explaining hypothetical events. *Journal of Expermental Social Psycholgy, 21*, 519-543.

Huys, J. Evers-Kiebooms, G. & d'Ydwalle, G. 1992. Decision making in the context of genetic risk: The use of scenarios. *Birth Defects:* Original Article Series. 28, 17-20

Klein G. 1998. *Sources of Power*. Cambridge, MA: The MIT Press.

Morris, R. & Ward, G. 2005. *The cognitive psychology of planning*. Hove, UK: Psychology Press.

Scaife, M., & Rogers, Y. 1996. External cognition: How do graphical representations work? *International Journal of Human-Computer Studies, 45,* 185-213.

Smith-Spark, J. H., Glasspool, D. W., Oettinger, A., Yule, P. and Fox, J. 2005. Planning, working memory, and interface support in a medical domain. In B. Hommel, G. Band, W. La Heij anf G. Wolters (Eds.) *Proceedings of the 14th Conference of the European Society for Cognitive Psychology.* Leiden, Netherlands: European Society for Cognitive Psychology. pp. 22-23.

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

Toulmin S. 1957. *The Uses of Argument.* Cambridge: Cambridge University Press.

Zhang, J., & Norman, D. A. 1994. Representations in distributed cognitive tasks. *Cognitive Science, 18,* 87-122.