Developing A Knowledge-based Tool for Authoring Clinical Trial Protocols

S Modgil[1], P Hammond[1], JC Wyatt[2], H Potts[2]
[1]*Biomedical Informatics Unit (Eastman Dental Institute for Oral Health Care Sciences)*
[2]*Knowledge Management Centre (School of Public Policy)*
University College London

Abstract. Many published clinical trials are poorly designed, suggesting that the protocol was incomplete, disorganised or contained errors. This fact, doctors' limited statistical skills and the shortage of medical statisticians, prompted the development of a knowledge-based aid, Design-a-Trial, for authoring clinical trial protocols. Design-a-Trial interviews a physician, prompts and guides them through suitable design options, comments on the statistical rigour and feasibility of their proposed design, and generates a 6-page draft protocol document. This paper reviews the progress of the Design-a-Trial project by describing a working prototype and its recent and planned development.

1 Introduction and Background

When designing a randomised controlled trial (RCT), the principal investigator (PI) writes a protocol document to justify the design and describe the trial procedures in detail. This protocol is used to obtain ethical approval for the trial, and is referred to by the principal investigator when the trial is written up for publication. Thus, it is a crucial document, and if incomplete, disorganised or incorrect, can prejudice the whole study. The complexity of trials from both a medical and statistical perspective makes protocol authoring difficult and time consuming. Moreover, clinicians and other health workers involved in authoring trials, often have limited experience in research and limited access to statisticians. Numerous studies have shown that bad design is all too common [2, 3, 16, 19]. This not only represents a waste of resources but also constitutes unethical behaviour towards patients who volunteer to take part in trials.

Design-a-Trial (DaT) is a knowledge-based decision support system that enables a user to produce more rigorous protocols more rapidly and more easily. The user describes a trial by entering data in forms. Expert knowledge encoded in the system is used to critique the design, so that the user is guided to design one that is scientifically and ethically sound. The system then generates a protocol document describing the designed trial.

While a number of software tools have been developed for management of clinical trials e.g., [1, 10], none of these support trial design. Unlike other trial design support tools, such as sample size calculators [14, 18] or systems like OPAL [11] (which allows a PI to describe a therapy plan of a trial, but assumes it is scientifically sound), DaT is intended as a complete system for use by those inexperienced in trial design, and contains the resources to help write draft protocols for many types of study from scratch. Recent work more closely related to DaT

includes [6, 15, 17], none of which provide the same scope of critiquing support together with the facility for protocol generation that DaT provides. These related works will be discussed in more detail in the conclusions.

We have developed two prototype versions of DaT. Version 1.1, developed in the early 1990s, ran under OS/2 in a mixed software environment and was described in detail in an earlier paper [20]. In order that DaT could be made available to a small group of users for ongoing evaluation while further development took place under new funding, DaT 1.1 was rationally reconstructed under Windows NT. The revised prototype, DaT 1.2, was completed in 1999 and is briefly described in Section 2. It is implemented entirely in the logic programming language Prolog [12] and provides an adaptive tab based interface, a comprehensive sample size calculator, a "critiquing" facility, and the automatic generation of a draft protocol document. Some preliminary evaluation of DaT 1.2 has taken place, suggesting further improvements and motivation for the ongoing development of the latest prototype DaT 2.0.

In Section 3, we describe ongoing work on DaT 2.0, including extension of DaT's expert knowledge base and development of a clinical trial ontology. Preliminary work on broadening the range of DaT outputs (clinical trial representations) is discussed. We describe a new user interface for DaT 2.0, focussing on an improved model of how users interact with the expert knowledge used for the generation of critiques. We also demonstrate one approach to coping with the unavoidable incompleteness of DaT's medical knowledge and the requirement for varying levels of detail in protocol documents.

2 The DaT 1.2 Prototype

DaT 1.2 recreates all of the functionality of its predecessor but contains a more extensive sample size calculator and a more readily negotiable interface to the data entry forms. Its knowledge base, now encoded in Prolog, is divided into a collection of medical facts (currently limited to thoracic medicine) and a set of expert rules. These expert rules have been encoded as constraints, and are used to generate critiques of methodological, medical, statistical and ethical aspects of clinical trial design. The knowledge base also contains definite clause grammars (*dcg*s) [12] used in the generation of the protocol and critique texts.

Figure 1: User interface in DaT 1.2

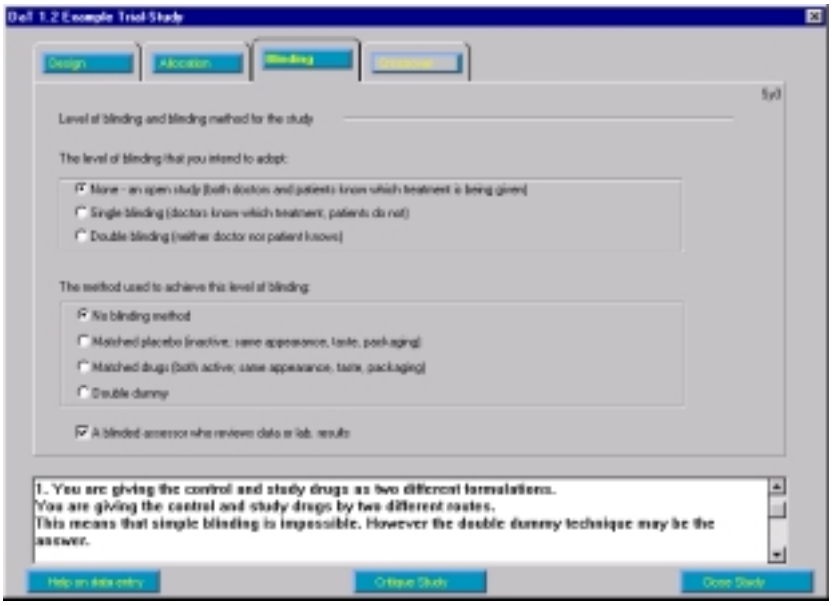


Figure 2: Example of a critique for the study section

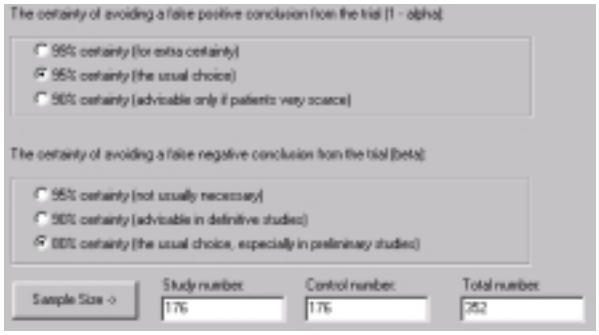


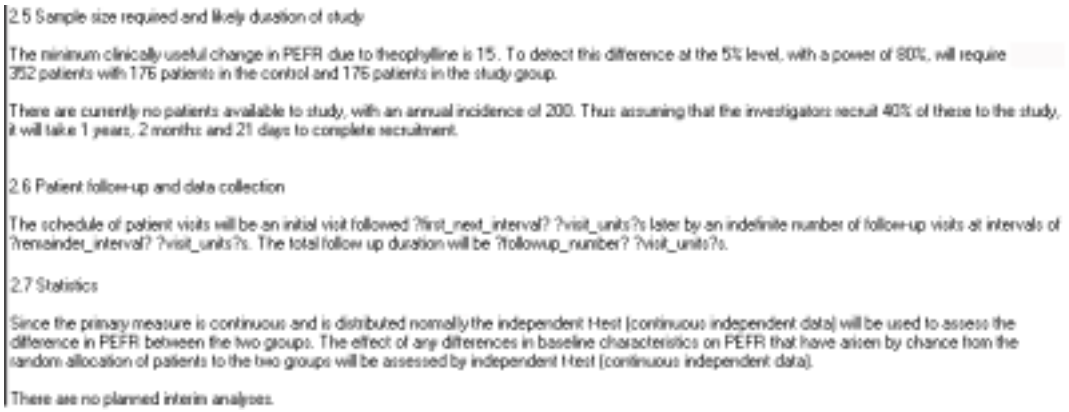
Figure 3: Example of a sample size calculation in DaT 1.2

The DaT 1.2 interface, also implemented in Prolog, employs a simple graphical representation of the components of a trial emphasising the typical order in which the main design subtasks should be undertaken (fig.1). A mouse click on any of the seven icons presents the user with a tabbed form interface opened at the appropriate set of forms.

Fig.2 shows the data entry forms for the “Study” section. If the user requests a critique having entered data relating to the control and study drugs in “Interventions”, and selected the blinding options shown, then the constraints relating to the “Study” section are checked. A contravention of a constraint is presented to the user as a *dcg* generated critique (bottom of fig.2).

A comprehensive set of sample size calculation routines have been written in the programming language C++. On the basis of data entered on the choice of design, allocation ratios, the type of primary measure, expected measure values, statistical test, and α and β values, the user can request a sample size calculation as shown in fig.3.

Once data entry is completed, the user is presented with options to edit the trial design, to view a concatenation of critiques applicable to, or generate the protocol document corre-



2.5 Sample size required and likely duration of study

The minimum clinically useful change in PEFR due to theophylline is 15. To detect this difference at the 5% level, with a power of 80%, will require 352 patients with 176 patients in the control and 176 patients in the study group.

There are currently no patients available to study, with an annual incidence of 200. Thus assuming that the investigators recruit 40% of these to the study, it will take 1 years, 2 months and 21 days to complete recruitment.

2.6 Patient follow-up and data collection

The schedule of patient visits will be an initial visit followed ?first_next_interval? ?visit_units?s later by an indefinite number of follow-up visits at intervals of ?remainder_interval? ?visit_units?s. The total follow up duration will be ?followup_number? ?visit_units?s.

2.7 Statistics

Since the primary measure is continuous and is distributed normally the independent t-test (continuous independent data) will be used to assess the difference in PEFR between the two groups. The effect of any differences in baseline characteristics on PEFR that have arisen by chance from the random allocation of patients to the two groups will be assessed by independent t-test (continuous independent data).

There are no planned interim analyses.

Figure 4: Selection of text from a protocol document generated for an incompletely specified trial

sponding to, the current state of the trial design. Selecting the last option presents the user with a 6 page protocol document. Part of an example protocol document is shown in fig.4. Notice that any missing data is indicated by meaningful (variable) names annotated with question marks.

3 DaT 2.0 in Development

In this section we discuss ongoing development of the latest version of Design-A-Trial: DaT 2.0. Fig. 5 may be referenced when reading the following sections.

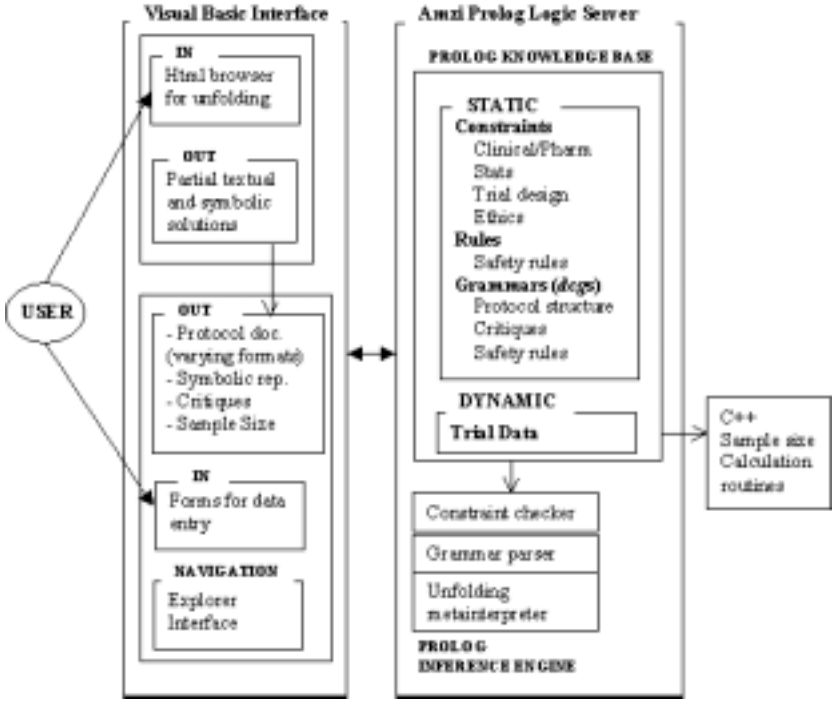


Figure 5: DaT 2.0 architecture

3.1 Knowledge Acquisition for DaT 2.0

DaT 1.2 supports the design of two group parallel and crossover trials, and the critiquing advice is limited to trials of drugs used in thoracic medicine. For DaT 2.0, we aim to increase the scope so as to cover a wider range of trial designs and medical specialities. Here, we outline some issues and strategies for these extensions of the expert knowledge base.

There is clinical data that may be useful for the purposes of trial design, but which is not readily available in a pre-existing form. For example, if DaT's knowledge base included the typical standard deviations seen with standard outcome measures, then this could be provided for power calculations. This would be a useful resource for clinicians who might not have such data to hand and would discourage users from underestimating standard deviations so as to reduce the result of the power calculation. We plan to obtain information on common diseases and standard outcome measures by approaching specialist societies in the UK. A questionnaire has been prepared and is currently being piloted. We will encourage an evidence-based approach in the compilation of such data.

Most of the statistical and methodological knowledge in DaT is well-established material, common to textbooks and courses aimed at the same audience as potential users. However, it became apparent that some of the rules being considered for DaT 2.0 were more contentious. For example, some statisticians have suggested that α and β should be set equal in power calculations. This is contrary to the usual practice of setting $\alpha = 5\%$ and $\beta = 1 - power = 20\%$. There is an ongoing debate about whether under-powered trials should be considered unethical or whether they are still of value to future systematic reviews and meta-analyses. We are conducting a survey of experts on such matters, which we hope will indicate areas of consensus and provide us with discussion material relating to contentious issues, for inclusion in the DaT 2.0 help files.

DaT uses pharmaceutical knowledge in a variety of ways. For example, it can use knowledge about contra-indications of a drug to deduce possible exclusion criteria for entry into a trial. Currently, pharmaceutical knowledge has been entered and stored as Prolog facts. However, since such knowledge is constantly evolving, one would want to establish intensional links with commercial drug databases. This aim will be discussed further in the conclusions.

3.2 An Ontology for Clinical Trials

Ontologies provide a means for structuring data, and can assist in specification of a system, helping to identify system requirements and to understand relationships among system components. We have developed an ontology for randomised controlled trials, and are currently coupling design of the DaT 2.0 interface with the ontology. We are also defining a mapping between the ontology and the Prolog symbolic schema that are instantiated by data entered when designing a trial. In this way, changes to the ontology can readily be integrated into the underlying symbolic representation of a trial. Finally, by providing a standard terminology for RCT concepts, our aim is to make the description of a designed RCT readily transportable to other clinical trial software. In this respect, our work can be viewed as a first step in the development of an Interlingua for clinical trials.

GLIF, the GuideLine Interchange Format [13], has proven to be a successful common format for software relating to guidelines. There are similarities between trials and guidelines which suggested that we base our ontology on GLIF. RCTs involve things being done

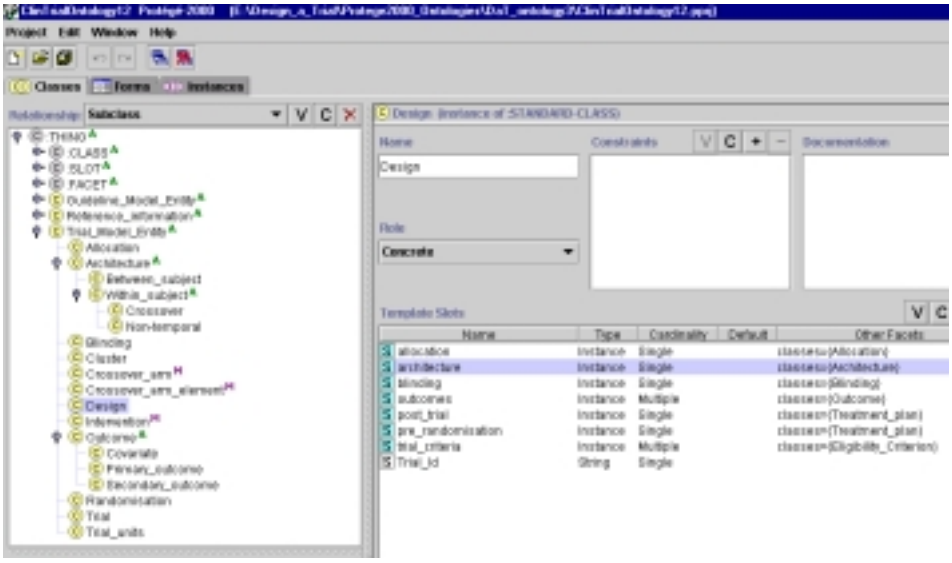


Figure 6: Screen shot from Protégé-2000 showing expansion of 'Trial_Model_Entity' and slots in the 'Design_subclass'

(interventions) to the units of the trial (usually patients). Guidelines, likewise, describe actions done to patients. Also, both RCTs and guidelines usually have inclusion and exclusion criteria.

GLIF was developed at Stanford Medical Informatics with their own Protégé software [5], which provides ontology editing functions. Our ontology is written using Protégé-2000 version 1.3.4. We took a GLIF 2 based draft ontology for breast cancer protocols as a starting point for our own work (this ontology was one among a number of example projects which came with an earlier version of the Protégé software). However, it became apparent that the requirements for a trial ontology meant a number of changes and extensions to the GLIF based ontology.

GLIF has a bipartite structure: one part describes the guideline, the other covers reference information. Fig. 6 shows the tree structure of the DaT ontology, which has a tripartite structure. The first part describes the interventions in a trial, and parallels the guideline description in GLIF. This is labelled 'Guideline_Model_Entity' in fig.6. Notice that not all RCTs are performed on patients. For example, one may carry out a trial of computer system use by clinicians. Thus, we have designed the ontology to be sufficiently generic to allow description of a wide range of trials, while concentrating on patient trials. This represents a departure from GLIF which is only concerned with the latter.

The second part covers reference information. Having a separate section for reference information allows for information to be referenced from other parts of the ontology. A particular measurement, for example, may be referenced as an outcome measure, as a safety check, and as an entry criterion.

The third part, labelled 'Trial_Model_Entity', is wholly new and describes the trial design. The ontology we have developed covers a variety of study designs, including cross-over studies, within-subject designs and cluster sampling. This section also covers: trial architecture; the process of allocation to different arms of the trial; blinding; outcome measures and associated statistics; and pre-randomisation procedures. Fig. 6 shows the 'Trial_Model_Entity'

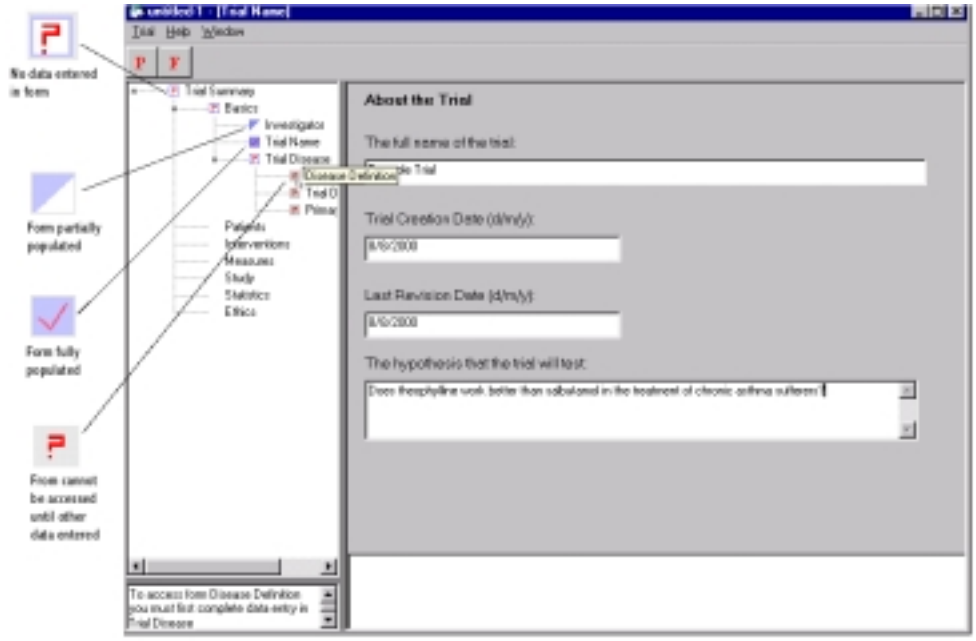


Figure 7: The experimental explorer style interface in DaT 2.0

section expanded to show all the children. Each subclass has many slots, which may refer to classes elsewhere in the ontology.

3.3 The DaT 2.0 User Interface

Experience gained in the development of DaT 1.2, and conclusions gleaned from its informal evaluation by clinicians and statisticians associated with the project, have suggested changes to the DaT 2.0 interface. Development of DaT 1.2 highlighted the impracticality of implementing a sophisticated graphical user interface in Prolog. On the other hand, Prolog is a natural choice for data representation, and particularly for the symbolic reasoning involved in constraint checking and natural language generation. This has motivated our use of Amzi Prolog [4] which enables Prolog programs to be compiled into a logic server module that communicates with other programming environments. DaT 2.0 currently has a Visual Basic interface, with data exchanged between the interface and an Amzi logic server module.

One problem noted with DaT 1.2 was the difficulty users experienced in orientating themselves in relation to the rest of the data entry forms. This suggested an explorer type navigation window containing a structured tree overview of the form hierarchy. Clicking on a node loads the appropriate form into the right hand window pane (fig.7). Associated with each node/form is an icon indicating the extent to which data entry has been completed for that form. The bottom left window provides advice on navigation.

3.4 Clinical Trial Representations

There are a number of orthogonal (although not necessarily isomorphic) representations of a trial. Firstly there is the protocol document itself. At any stage during data entry in DaT 2.0, the user can request a draft protocol document reflecting the current state of completion of the

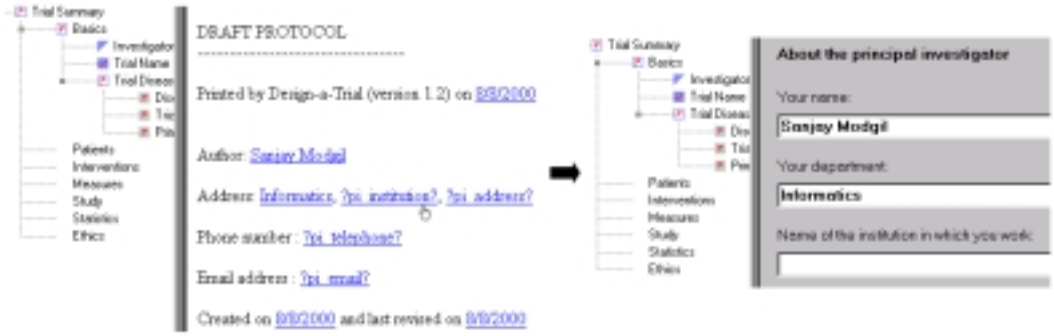


Figure 8: Browser format of an incomplete protocol document with pointers to form entry fields

trial description. The text is generated by a definite clause grammar which at present can be parameterised to generate the protocol in either ASCII text or the markup language HTML. We also intend parameterising the grammar so as to increase both the range of text formats generated, and protocol formats (in terms of text arrangement and content) complying with the requirements of different specialist societies and the needs of different regulatory bodies.

Fig.8 shows the HTML format of an incomplete protocol document in a web style browser window. Notice the hyperlinks associated with data acquired from the data entry forms, as well as data required but not entered and so parenthesised by question marks. Clicking on these links returns the user to the form where that particular data item was (or needs to be) entered. Fig.8 shows a user clicking on the linked text ?pi_institution?, whereupon the form entitled “Investigator” is displayed and the user can respond to the prompt “Name of the institution in which you work”. This functionality has been implemented in response to difficulties experienced by DaT 1.2 users when locating the appropriate form for entering data identified as missing in the protocol text.

Another trial representation is the underlying symbolic Prolog representation of a designed trial. We intend formalising mappings of this representation to formats suitable for export to other trial software (in particular trial management software such as [10]). For example, we have presently implemented a mapping to the CLIPS format used by Protégé-2000, so that a DaT designed trial may be viewed as an instance of the clinical trial ontology in the Protégé-2000 software.

3.5 Pre-empting Violation of Constraints

In section 2 we described how critiques local to each task design subtask may be generated during data entry. However, this critiquing model of constraint animation may lead to problems of “reason maintenance”, i.e., having to revise data in response to a critique may have knock on effects on other data entered. To illustrate, a critique in the ethics section of a DaT 1.2 example trial suggests that the user justify their choice of theophyline as the control drug, given that there are other asthma drugs with fewer side-effects. Such justification is required for ethical approval. The critique also suggests alternative choices for the control drug (those with fewest side-effects). The user may be prompted by the critique to change the choice of theophyline as the control drug. This may subsequently require changes to other data previously entered on the basis of having chosen theophyline. This problem has suggested a

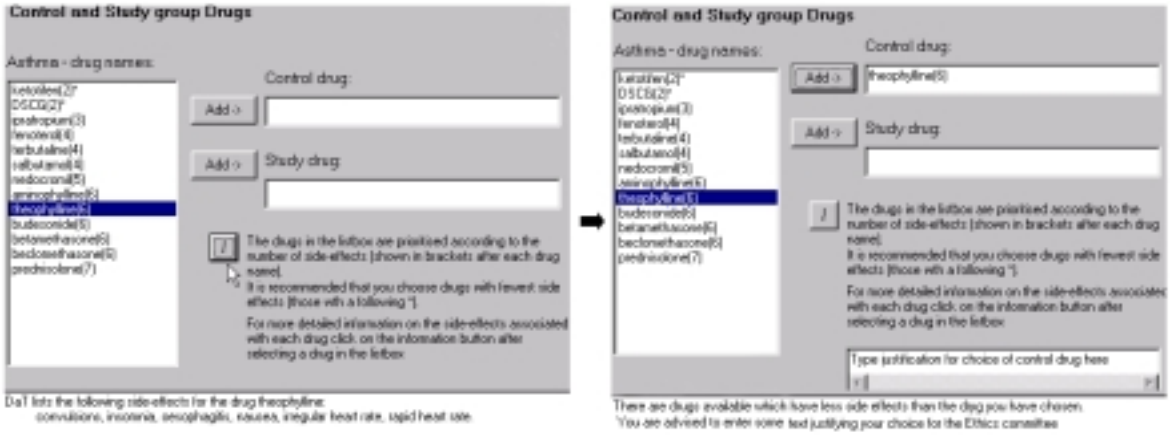


Figure 9: Users selection of a drug not of highest rank results in a request from DaT for a justification

rethink of how such constraint knowledge should be animated in DaT 2.0:

Where possible, critiques should be pre-empted; by encouraging the user, from the outset, to enter data that conforms to the constraints.

This approach has motivated changes to the way information is displayed to and elicited from the user. To illustrate, consider the pre-emption of constraint violation relating to the choice of control drug (see fig.9). When choosing asthma as the disease for investigation, the user is presented with a prioritised list of drugs from which to choose the control drug. If requested, DaT will expand on the rationale for the prioritisation (i.e., showing the side-effects)[1]. The user can then choose from among the drugs with highest ranking. If the user doesn't do this, then DaT generates a request for a justification to be made. The supplied justification can then be included in the draft of the protocol document.

3.6 Generating protocol text descriptions concerning adverse events

Adverse events are frequently given a high profile in protocol documents, especially in those trials involving chemotherapeutic agents. In this section, we demonstrate how generic principles for managing adverse events might be used to generate textual extracts for inclusion in the protocol, even when DaT's knowledge base has incomplete information about an intervention, its side-effects or its mode of administration.

Previously, ten generic safety principles were abstracted from an empirical analysis of chemotherapy protocols [7] (see table 1). These principles, when encoded as obligations, can be used to generate advice as to how a clinical trial should be further specified, or indeed revised, to conform to these obligations. To illustrate, the Prolog representation of the *prevention* principle (specialised to an extent) is shown below:

[1]Admittedly this is a somewhat impoverished basis for prioritisation. We anticipate making use of a richer set of criteria, as well as more complex algorithms for determining a ranking.

Table 1: generic safety principles

Exacerbation	Avoid exacerbating anticipated hazards	**Monitoring**	Monitor responses which herald hazardous situations
Diminution	Avoid undermining the benefits of essential actions	**Efficacy**	Ensure that overall plans are efficacious in pursuing stated objectives
Interaction	Ensure that an action does not interact with another action to cause harm	**Sequencing**	Order (essential) actions temporally for good effect and least harm
Reaction	React appropriately to ameliorate detected hazards	**Critiquing**	Critique proposal of certain hazardous actions even if they are well motivated
Warning	Warn about hazards due to incorrect execution of essential actions	**Prevention**	Prevent or ameliorate hazards before executing an essential action

$$advise\_plan(Action, Plan, Prevention, ameliorate(Action, Effect)) :- \quad (1)$$
$$part\_of\_plan(Action, Plan),$$
$$effect(Action, Effect),$$
$$hazardous(Effect),$$
$$prevention(Action, Effect, Prevention).$$

This rule (the head of the rule) advises inclusion of preventative measures $Prevention$ in a $Plan$ (clinical trial), if ($:-$) the following goals in the body of the rule are met: some $Action$ is part of the $Plan$; that $Action$ has an $Effect$; the $Effect$ is hazardous; and the $Effect$ is ameliorated by $Prevention$.

Suppose we have partially specified a clinical trial which includes an action to administer a chemotherapeutic drug $cisplatin$. Suppose also that the expert knowledge base contains (1), and information about hazardous effects of $cisplatin$ and preventative measures to ameliorate these effects. In DaT 2.0, the user will be prompted (via a suitable interface) to execute the query:

$$?- advise\_plan(administer(cisplatin), plan\_1, Prevention, \quad (2)$$
$$ameliorate(administer(cisplatin), Effect)).$$

which might return the following solution:

$$advise\_plan(administer(cisplatin), plan\_1, \quad (3)$$
$$[pre(administer(cisplatin), hydration(default, [normal\_saline, 1-2, litre])),$$
$$post(administer(cisplatin), hydration(default, [normal\_saline, 1-2, litre]))],$$
$$ameliorate(administer(cisplatin), dehydration)).$$

prompting the user to confirm inclusion of the pre- and post- hydration regimes in the symbolic representation of the trial. Corresponding to (3), one would want to include the following textual clause in the protocol:

> Preventative actions: (4)
> - prior to administering cisplatin administer the default regime: normal saline 1-2 litre
> - after administering cisplatin administer the default regime: normal saline 1-2 litre
> are advised in plan 1 (administering cisplatin causes dehydration).

However, the above model suffers a number of drawbacks:

- Evaluating a query on an incomplete knowledge base may result in failure, ignoring the possibility of partial and informative solutions. For example, the query (2) may fail if details of hydration regimes are not included in the knowledge base. This ignores the following useful information that could be derived during query evaluation:

> Preventative actions: (5)
> - administer a type of hydration regime prior to administering cisplatin
> - administer a type of hydration regime after administering cisplatin
> are advised in plan 1 (administering cisplatin causes dehydration).

  The problem of incompleteness of the knowledge base is particularly pertinent, given that clinical trials often assess new drugs as well as unusual combinations of drugs. Inclusion in the protocol of textual clauses such as (5), not only ensures safe enactment of the trial, but also communicates awareness of hazards and the need for remedial actions to the regulatory bodies which sanction trials. Furthermore, one can include the underlying symbolic representation of clauses such as (5) in the trial specification to be exported to software tools for enacting/managing trials (should the knowledge base be updated later, one can then attempt to complete evaluation of the partial symbolic solution corresponding to (5)).

- In contrast to the above, the user may be overwhelmed with a surplus of information. For example, the user may be presented with a long list of adverse effects of cisplatin, together with preventative measures, despite the fact that some of these adverse effects may be rare, and ordinarily would not need to be accounted for.

- A problem related to the above is the lack of control over the degree of detail of the advice given. For example, for some protocols (and indeed some adverse effects) it may be sufficient to simply state the requirement for taking preventative measures, rather than specify the details of these measures. Also, different regulatory bodies require varying levels of detail in the protocol.

To address the above drawbacks we have implemented a meta-interpreter for query evaluation [8, 9]. The user interacts with the meta-interpreter via natural language (NL) translations of Prolog code. The user is initially presented with an NL representation of a rule such as (1). In a process we term ”unfolding”, the user selects from NL representations of goals in the

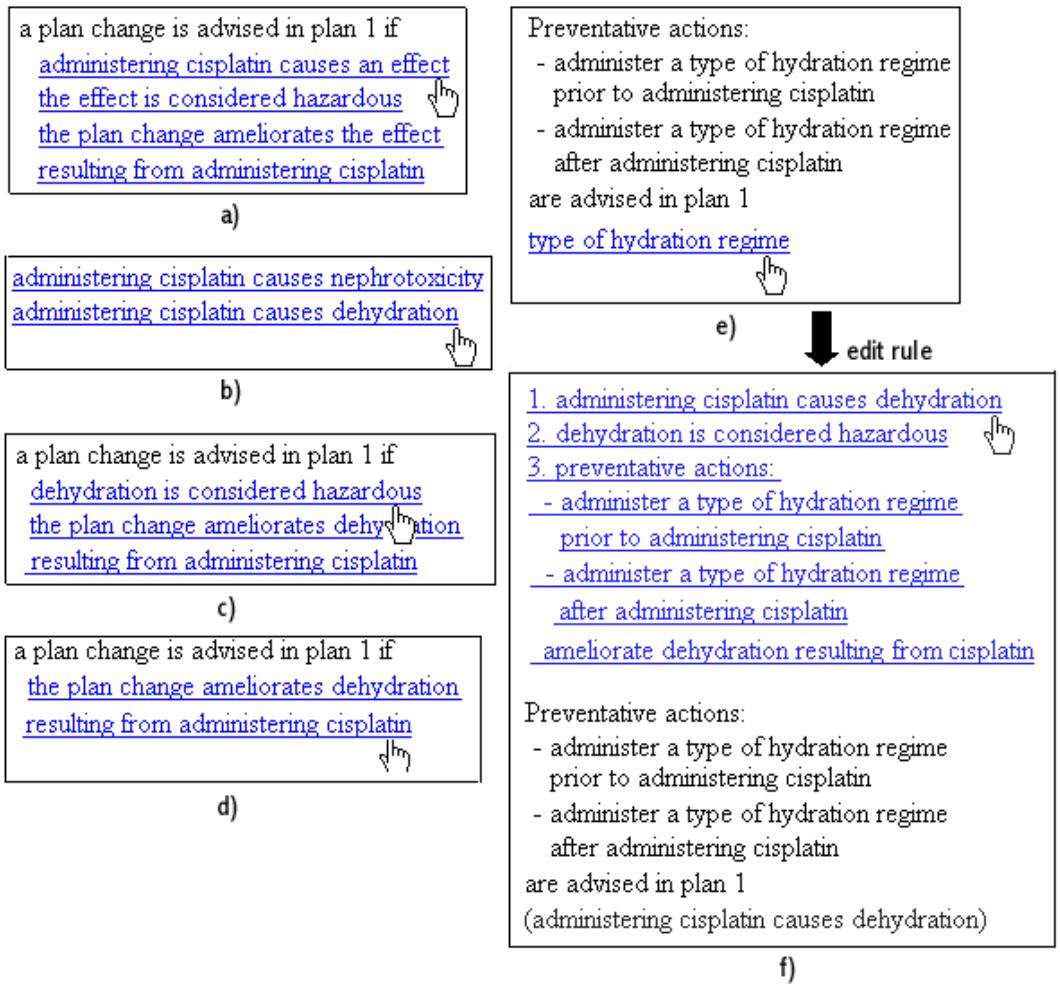


Figure 10: Screen shots showing user unfolding prevention principle via natural language translations of prolog code

body of the rule. Subsequently, NL versions of those clauses in the knowledge base whose head matches (resolves with) the selected goal are shown. The user selects from among the clauses, which results in further specialisation of the rule. The specialised rule is then presented for further unfolding.

The process described gives the user full control over the path taken through the search space of solutions. Once a goal fails because of incomplete knowledge, the user can include a symbolic representation of the partial solution (thus far obtained in the unfolding process) in the trial, as well as export the corresponding textual clause (e.g., (5)) to the protocol. Also, control over navigation though the search space means that the user can avoid generation of esoteric solutions, and also unfold to the required level of detail. We illustrate with an example. The following description of how (5) is generated should be read with reference to fig.10.

Suppose a knowledge base containing (1), and the following Prolog clauses:

$$part\_of\_plan(administer(cisplatin)). \quad (6)$$

$$effect(administer(cisplatin), dehydration). \quad (7)$$

$$effect(administer(cisplatin), nephrotoxicity). \quad (8)$$

$$hazardous(dehydration). \quad (9)$$

$$prevention(administer(cisplatin), dehydration, [pre(administer(cisplatin), hydration(Type, P)), post(administer(cisplatin), hydration(Type, P))]) :- hydration(Type, P). \quad (10)$$

Initially, a dialog presents a choice of actions previously specified as part of the plan, and a list of principles to choose from. Selecting administration of cisplatin as part of plan 1, and the *prevention* principle, the first goal in the body of (1) is automatically unfolded. The resultant rule and the goals in its body are displayed in a web-browser window (fig.10a). The user can now unfold the rule. Unfolding the first goal, the matching clauses (7) and (8) are found, translated and displayed (fig.10b). Selecting (7), then the resultant new rule and the goals in its body are translated and displayed (fig.10c). Choosing the goal "dehydration is considered hazardous" then since there is only one matching clause (9) in the knowledge base, the goal is automatically unfolded (fig.10d). Choosing the only available goal, then as in the previous step, there is only one available clause (10). So, automatic unfolding gives the result shown in fig.10e. When selecting "type of hydration regime" the user is informed that there are no clauses describing types of hydration regimes in the knowledge base. The user chooses to edit the rule thus far obtained, and is shown a history of the heads of clauses selected (including those unique clauses not shown to the user for selection). These can then be selected for amending in parentheses to the textual rule as shown in fig.10f. The resulting text can be exported to the protocol, and the underlying symbolic partial solution included in the symbolic representation of the trial. Notice that the user has full control over the number and kinds of solutions generated, and can control the level of detail required, deciding to cease unfolding at any stage.

4 Conclusions

In this paper, we have advocated the need for software support for clinical trial design, and described the development of a prototype software system - Design-a-Trial - which supports data entry, generates critiques, supports statistical calculations, and generates a protocol document. We briefly described the completed prototype DaT 1.2, implemented in Prolog, and described current work on the latest prototype DaT 2.0. We motivated and described an experimental Visual Basic interface for DaT 2.0, with use of Prolog restricted to its traditional strengths; knowledge representation, symbolic reasoning and natural language processing.

We described our specification of a clinical trial ontology which provides a data model for DaT 2.0 and will hopefully serve as an Interlingua for clinical trials, enabling communication of trial descriptions among different clinical trial software. Plans for extending the scope of DaT 1.2 have been described, with the aim of integrating expert knowledge so that DaT 2.0 covers a wider range of medical fields and types of trial design. We have felt it more appropriate to explore linkage of DaT 2.0 to commercial drug databases in the subsequent commercialisation of DaT (negotiations for combined government and commercial funding

are underway). This is because associated negotiations over intellectual property rights are less appropriate in a research project. We discussed extending the range of formats of clinical trial representations (protocols and symbolic representations) generated by DaT. Future work will also focus on graphical representations of clinical trial time lines, which are often included in protocols.

For DaT 2.0, we motivated and illustrated a change in the way expert knowledge, currently encoded as constraints, should be animated so as to guide sound and good trial design: i.e., pre-empting constraint violation rather than critiquing post constraint violation. We also described a novel approach to the generation of partial symbolic solutions and their corresponding textual representations (for inclusion in a protocol) upon requesting advice relating to safety aspects of treatment plans. We showed that this approach can ameliorate the inevitable incompleteness of medical knowledge bases, and meet the requirements for generating solutions of varying degrees of detail. Future work will apply this approach to the specification of selection criteria for a clinical trial. We are currently working on abstracting general principles (obligations) relating to selection criteria. As was described for the safety principles, we aim to provide a natural language interface for unfolding of these principles.

Once completed, DaT 2.0 will be assessed in a blinded randomised experiment to evaluate its impact on protocol quality. Principal investigators (PIs) will be randomised into two groups: a group using DaT 2 versus a control group. PIs in each group will write a protocol, with the outcome measure being a score of protocol quality.

We conclude by mentioning work closely related to our own. The aims of the PaTriCIa project [6] are similar to those of DaT. However, support for statistical calculation and extensive critiquing is not provided. [6] describes an object orientated knowledge representation model underlying a dynamic user interface for data entry. Associated with each object type is a section of protocol text with variables appropriately instantiated upon data entry. Expert knowledge is encoded in relationships between object types. This knowledge is exploited in a similar manner to our proposal for pre-emption of critique violation described in section 3.5. However, the relationships described in [6] appear to be limited to statistical design aspects of clinical trials.

A company [15] has developed software which allows aspects of a trial to be described, but excludes critiques of medical and ethical aspects. Neither is there support for the generation of protocol text. However, [15] does provide considerable support for optimising the statistical aspects of a trial, by allowing participants to test multiple ”what if” assumptions. For example, the power of a trial to show a difference between treatment arms can be compared across scenarios (virtual instances of the described trial) to compare the impact of drug dose strategy.

Finally, [17] describes early development of a protocol-critiquing tool. The critiques cover medical and procedural content, such as patient treatments, monitoring tests, and drug toxicity, but do not address the statistical, design and ethical content of a trial.

References

[1] ACS; ” http://www.acs-world.com/.

[2] Altman D.G. Statistics in medical journals. Stat Med 1983;1:59-71.

[3] Altman D, Dore C. Randomisation and baseline comparisons in clinical trials. In: Lancet 1990; 335: 149-153.

[4] Amzi; www.amzi.com.

[5] Grosso W.E., Eriksson H., Fergerson R.W., Gennari J.H., Tu S.W., and Musen M.A. Knowledge Modelling at the Millennium (The Design and Evolution of Protégé-2000) 1999. SMI Report Number: SMI-1999-0801.

[6] Haag U. Knowledge representation for computer-aided planning of controlled clinical trials: the PaTriCIa project. In Methods of Information in Medicine. 1997; 36(3): 172-178.

[7] Hammond P., Harris A.L., Das S.K., and Wyatt J.C. Safety and decision support in oncology. In: Methods of Information in Medicine. 1994; 33(4): 371-381.

[8] Hammond P., Sergot M.J., and Wyatt J.C. Safety reasoning in medical decision support. In: Artificial Intelligence in Medicine: Research Frontiers and Funding Opportunities. I.E.E. Colloquium, Savoy Place, London, U.K. (1st.). 1996; I.E.E. Digest No: 96/031.

[9] Hammond P., Modgil S., and Wyatt J.C. Safety and Computer-Aided Design of Chemotherapy Plans. In: Topics in Health Information Management 2000; 20(4): 55-67.

[10] MACRO; www.infermed.co.uk/macro_tm.html.

[11] Musen M. Automated generation of model-based knowledge acquisition tools. London: Pitman 1989.

[12] Nilsson U., and Matuszynski J. Logic, Programming and Prolog. John Wiley & Sons 1990.

[13] Ohno-Machado L., Gennari J.H., Murphy S., Jain N.L., Tu S.W., Oliver, D.E., et al. The GuideLine Interchange Format: A Model for Representing Guidelines. In: Journal of the American Medical Informatics Association, 1998; 5(4): 357-372.

[14] PASS 2000 - Power analysis and sample size calculator; www.ncss.com/pass.html.

[15] Pharsight; www.pharsight.com

[16] Pocock S.J, Hughes M.D, Lee R.J. Statistical problems in the reporting of clinical trials. A survey of three medical journals. In: N Engl J Med 1987; 317:426-32. [Medline].

[17] Rubin D.L, Gennari J, & Musen M.A. Knowledge Representation and Tool Support for Critiquing Clinical Trial Protocols. SMI Report Number: SMI-2000-0844.

[18] SISA - Simple Interactive Statistical Analysis; http://home.clara.net/sisa/about.htm.

[19] Smith D.G., Clemens J., Crede W., Harvey M., Gracely E.J. Impact of multiple comparisons in randomised clinical trials. In: Am J Med 1987;83:545-50.

[20] Wyatt J., Altman D., Heathfield H., Pantin C. Development of Design-a-Trial, a knowledge-based critiquing system for authors of clinical trial protocols. In: Comp Prog Meth Biomed 1994; 43: 283-291.