A Cancer protocol writer's assistant

by

Brij Mohan Masand

B. Tech., Indian Institute of Technology, New Delhi.

SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE

DEGREE OF

MASTER OF SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Summer, 1982

Signature of Author _____
Dept. of Electrical Engineering and Computer Science
Summer, 1982

Certified by _____
Peter Szolovits, Associate Prof., Dept. of Electrical Engg. and Computer Science
Thesis Supervisor

Accepted by _____
Chairman, Departmental Graduate Committee

A Cancer protocol writer's assistant

by

Brij Mohan Masand

Submitted to the Department of Electrical Engineering and Computer Science
on June 24, 1982 in partial fulfillment of the
requirements for the Degree of Master Of Science
in Computer Science

ABSTRACT

Comparisons of different treatments are carried out by conducting clinical trials of the treatments. A protocol is a plan for conducting a clinical trial. Many cancer protocols suffer from problems of ambiguity, incompleteness inconsistensy and inadequate statistical design in specifying the clinical trial. The design and implementation of an interactive program is described which can help physicians write better protocols. The program has the representation of the formal structure of a protocol and the disease related knowledge in the specific domain of lung cancer. It uses the constraints in the structure of the protocol and the disease related knowledge to detect incomplete or inconsistent specifications of the clinical trial. A separate statistical advisor helps in the statistical design of the protocol. Under the scope of this project the program deals with protocols involving chemotherapy only. The program can be used by doctors unfamilar with computers.

Keywords: Expert system, Computers in therapy design, Artificial Intelligence in medicine, Cancer Protocols.

Thesis supervisor: Peter Szolovits

Title: Associate Professor of Electrical Engineering and Computer Science

Acknowledgements

I would like to thank all those who directly or indirectly made this thesis possible. In particular I would like to thank

Peter Szolovits, who put up with me, for his support and direction at every stage of the project, for the continuous stream of ideas that he provided, for the excellent ways of identifying and explaining central ideas in different AI techniques, for organizing the project and clarifying the issues concerned, for reading my thesis in a most detailed manner, suggesting improvements for both content and style and finally for maintaining a lively, friendly and cohesive research atmosphere within the group.

Bill Long who was always available to answer my questions and always patiently listened and corrected my nebulous ideas, for helping me to focus on specific issues and goals, especially in the early stages of the project.

Drs. R. Friedman and J. Krikorian for their contributions to this project, both for getting it off the ground in terms of providing the knowledge as also for helping evaluate the goals, progress and scope of the project.

Bill Martin for introducing me to the wonderful world of representation and natural language understanding.

Ramesh Patil for being an excellent sounding board for ideas and for "debugging" and clarifying many of my fuzzy ideas, for the many incisive constructive criticisms regarding different aspects of the the project and helping me formulate and identify the relevant issues precisely, for teaching me to be more accurate in expressing myself and for the many tips for using the system more efficiently.

Glenn Burke for the constant help, information and programming ideas that he provided, for making LISP and ITS less mysterious to me, for being the "wizard" to run to for help when anything "broke"!

Harold Goldberger for the many discussions and advice about everything under the sun!

Lowell Hawkinson and Rajeev for many illuminating discussions.

Howard Sherman, Ken Church, Tom Russ, Irwin Asbell, Chris Eliot, Rosie and other past and present members of the MEDG group, from whom I benefitted through discussions and who have made MEDG such a friendly, supportive and fun group to be in and to work with.

Nafisa, Sriram, Billy, Lisa, Peter, Michele, Ann, Connie, Al and all my other friends for being understanding and supportive.

My mother and sisters for bearing the hardships caused by my being away from them.

CONTENTS

FIGURES

1. INTRODUCTION

Most of the AI programs in medicine, such as the INTERNIST, PIP, CASNET, MYCIN and others have concentrated on diagnosis as a problem and only a few such as the DIGITALIS ADVISOR and ONCOCYN among others, have dealt with the domain of therapy. This project is concerned with therapy, in particular the efficient *design* of new chemotherapy in the domain of lung cancer. It falls in the general class of knowledge based systems.

The aim of this project was to develop a working program that would help doctors in designing new clinical trials of chemotherapy. More specifically the aim was to develop an interactive program which will help doctors write a complete, consistent, statistically sound and unambiguous specification of a clinical trial. Such an interactive program has been developed. The program can assist a doctor design and express a new protocol for lung cancer.

The eliciting and formalization of the structure of the protocol and its related knowledge has been an important activity of the project and has built the foundation on which much more can be attempted. At this stage of the project we have more clear distinctions between the desirable, the interesting, and the feasible . The important issues have been brought to light, even if ways to deal with all of them haven't yet been found. Apart from the issues specific to the domain we have dealt with the issues that arise in a practical project of this kind. These issues include, among others, problems of representation of knowledge (in cancer and in general), of the availability and formalizablity of such knowledge, computational problems, the appropriateness of different AI techniques and user engineering.

This chapter describes the general background, specifies the problem, outlines the approach, and briefly refers to the problems encountered and the lessons learnt.

Chapter 2 is a demonstration of the system which gives an idea of what the system can do and what are its limitations. It is an annotated excerpt of the design of a particular simplified protocol.

Chapter 3 dives into the heart of the problem describing the structure of the protocol and sets up the detailed background in terms of which to illustrate specific issues and solutions.

Chapter 4 describes the organization of the program, both the representation and the control structures, also commenting briefly on why certain choices were made.

Chapter 5 examines in detail some of the important computational problems encountered. Some of them derive from the AI issues involved, others from a pure computer science point of view. They include problems of naming, instantiating structures, storing, displaying protocols, aborting and continuing etc.

Chapter 6 recounts the successes, shortcomings and the lessons of this project in some detail. Among other things it looks at the domain of cancer therapy and the appropriateness of AI techniques to it, and with the hindsight that is so abundant at the conclusion of a project, describes suggestions for future directions.

1.1 Background

Cancer accounts for about 400,000 deaths annually in the U.S. out of which about 100,000 are caused by lung cancer. Advances in the treatment of lung cancer, which may involve chemotherapy, surgery and radiation, are quite infrequent and even today, from the time of diagnosis the average survival is 6-9 months with less than 20% surviving more than 1 year.

In the absence of any "miracle drug" most advances rely on more mundane approaches such as improvements in the traditional treatments, in the form of new combinations of known agents and variations in the therapy itself. These treatments may be multi-modal i.e. may involve combinations of chemotherapy, surgery and radiation therapy, although for the purpose of this project we will be looking at protocols involving chemotherapy only.

To test and compare different treatments - which may be variations of old ones or new treatments - experiments in the form of clinical trials are carried out. They occur in three phases (1) Toxicologic testing to determine tolerable doses, optimal routes and timing of new anti-cancer agents. (2) Trials of the agents to determine their spectrum of effectiveness. (3) Comparison of effective drugs and their combinations to develop optimal treatment strategies. These trials are guided by detailed descriptions of the study called a "Protocol". Due to the large number of patients required to assure statistical significance of the result, these studies are typically carried out simultaneously at several institutions. The protocols serve to standardize and coordinate the therapy. They typically contain instructions for initial patient selection, detailed description of the treatment, instructions to deal with undesirable side effects, definitions of response criteria and

study parameters. These protocols essentially provide a specific approach to the disease so that the treatment is administered in a uniform way and reliable statistical data is generated over a period of time for comparing the relative efficacy of different treatments.

1.2 The General Problem

The success of a clinical trial depends on three broad categories of factors

The design of the therapy in terms of a good higher level rationale

The information gathering part of the study which relies on a good statistical design, a complete specification of the therapy and compliance on the part of the participating physicians.

The analysis of results.

If AI techniques are to be applied to this general domain of *designing* a protocol it would mean that it would need expertise in the first two areas mentioned above. The design of the therapy itself is a very complex process involving empirical and heuristic knowledge of the the spectrum and efficacy of drugs and, for reasons described in detail later, is beyond the scope of a formalized implementation in the form of a computer program. Due to the state of the knowledge in the field it is not possible to envisage at least in the near future programs which will help optimize therapy from their knowledge of mechanisms of drugs and disease processes. In this project we provide a framework to design a protocol which provides a facility to express the complete protocol and refine the information gathering part of the clinical trial, which consists of :

The statistical design of the protocol

The actual description of the therapy which should be complete, consistent and unambiguous in order to be administered appropriately. To a large extent this overlaps with the design of the therapy itself because they are so closely linked together, which also limits its "intelligence", in the absence of a higher level knowledge.

1.3 The Specific Problem

With the increase in clinical trials there has been a proliferation of protocols and there are about 1,500 of these trials in progress all over the world.

Unfortunately many of these protocols suffer from serious problems involving ambiguous, incomplete or inconsistent specification of treatment procedures as has been shown by Szolovits *et al* [Szolovits79]. For instance a certain rule for modifying drug dosage in case of toxicity might say "If the WBC count goes below 4000/cubic-mm then reduce the dosage of Methotrexate to 75%" this rule doesn't specify whether the new dosage is 75% of the original dose or 75% of the last dose. Another example is the rule "Withdraw the drug Vincristine in case of severe neuro-toxicity" without specifying how to judge the severity of the toxicity. Other errors in the protocol are contradictory instructions or serious omissions. For instance the eligibility section may fail to screen patients for heart disease when it is known that a certain drug will cause cardiac toxicity during treatment or the protocol may fail to specify when the treatment terminates. Further many of the protocols may be of poor statistical design, making it less probable that true results will be detected. Missing or ambiguous specifications force the physician to take individual decisions and this not only potentially reduces the efficacy of the treatment but prevents reliable statistical data from being gathered because of the non-uniform interpretations of the protocol by different physicians. The aim of this project is to develop an interactive program which will help a doctor write a complete, consistent, unambiguous and statistically sound protocol. To be able to do this the program must have knowledge regarding the structure of the protocol, knowledge about drugs and disease and the associated constraints so that it can apply the constraints to check for incompleteness, ambiguity and inconsistency. Under the scope of this project, the program is restricted to dealing with protocols involving chemotherapy only.

Some of the possible areas in which expertise is needed, are:

A facility to just *express* the protocol unambiguously and consistently. This will require knowledge of the structure of the protocol, knowledge about the cancer, the drugs, and toxicities.

Drug process simulation: This can enable the doctor to select the best synergistic combination from a number of them.

Simulation of the toxicities: In practice it is the ratio of the therapeutic to the toxic effect that we are more concerned with rather than just the therapeutic effect alone. Thus the goal is to minimize toxicity for a given level of therapeutic effect or to maximize the therapeutic effect for a given a level of toxicity. A simulation of toxicities helps to see the effect of different drugs at the same time.

Analytical, statistical and logical design of the protocol: This would enable the doctor to see if the study is feasible on statistical grounds at all and will help to maximize the information that he can gather from available resources. This is a much needed area of expertise for which usually a statistician is called in.

A demographic/epidemiological study to estimate the number of patients accruing: This can be an adjunct to the statistical section of the protocol where once the number of patients required are known, it can determine whether the given patients population with its given frequency statistics for different age groups will yield the required number of patients in the time limits of the study.

Help in designing new combinations of drugs/strategies from a knowledge of individual drug processes and cell kinetics etc.

Although it would be ideal to have a program which would deal with all of the above, in practice the goals have to be very limited to be able to have a feasible, working program, in the light of problems of the availability, formalizability of knowledge and the complexity of some of the above problems. The scope of this project is limited to having a facility to express the protocol consistently and unambiguously, be able to design the statistical part of the protocol, and have an elementary simulation of toxicities to help schedule the drugs better. The simulation of drug processes has been judged to be infeasible at this time due to inadequate knowledge about drug processes and the inevitable complexity of such knowledge when it is available. A more detailed discussion of the domain of cancer and the appropriateness of AI techniques to it appears in the conclusion.

It would seem that all that is required is to formalize the structure of the protocol and expose the constraints so that it is easy to spot the missing or inconsistent specifications. While this is true in principle there are problems which arise from incomplete knowledge of cancer, drugs (their therapeutic and toxic effects), complexity of descriptive specifications and the variations in the protocols themselves. Along with these basic issues there are requirements relating to the user engineering aspects of the project. The structure of the program has to conform to needs such as having a facility to modify the protocol --in a data-base with constraints,

this is a nontrivial problem, and the I/O has to be "friendly" to the user. Before describing our general approach we briefly review related work.

1.4 Related work by others

Thomas Russ at MIT completed his bachelor's thesis on designing a preliminary program for cancer therapy design project dealing with the treatment section of the protocol. His work has been very useful in getting an insight into the kinds of problems involved and also in organizing the knowledge required in the treatment section (e.g. drug related knowledge, parameter related knowledge and information about toxicities.)

At Stanford Shortliffe *et al* are developing a system called ONCOCIN for protocol management [Shortliffe81]. The project is specifically intended to manage *existing protocols* and is not oriented towards designing new ones. The protocols are represented in a rule based scheme derived from MYCIN and augmented by a data driven system. It is designed to make the existing protocols and their rules easily accessible to the physicians. Its primary aim is to make the *management* of existing cancer therapy easier. It can also advise physicians regarding possible therapeutic options.

Other works are related more indirectly. Mike Bosyj developed a program called PROCTOR to aid development of a procurement system. The program has knowledge of a general procurement system and based on the user's response to questions it can generate a customized procurement system. This is qualitatively similar to this project in the sense that constraints are used to correlate facts and changes in different parts of the system. The only difference is that PROCTOR assumes that the user doesn't have a ready design while in this project it is assumed that the doctor already has a fairly specific protocol design and the program allows him to express the design and perform relevant checks on the design.

Jon Doyle's research concerns Truth Maintainance Systems (TMS) [Doyle79]. It is a system based on non monotonic logic and uses dependency relations to "justify" deduction and to detect contradictions. This system was found to have many problems regarding it's adaptation to detect the kind of contradictions involved in the cancer domain. The problems mainly originate from representing "world knowledge" in a framework suitable for TMS. For instance representing general truths such as "For all drugs there exists ...." leads to problems of large blowup in the

TMS system. It does suggest a framework to represent protocol violations as contradictions to be detected and resolved in the typical TMS fashion -- by questioning the offending assumptions and retracting them in the order of least importance.

1.5 Basic approach to the problem

The initial part of the project was devoted to studying various protocols and formalizing their structure. In collaboration with Oncologists at the BU Hospital, knowledge about the disease and constraints within the protocol were elicited and represented within the program data-base. Having this generic model of the structure of the protocol, the program can help a doctor express a new protocol. While the protocol is being designed in an interactive, conversational mode, the program checks the incoming data for possible errors. When parts of the protocol are complete it then checks them for internal (local) constraints and later it checks (global) consistency with other parts of the protocol. To illustrate this, consider the interaction when the doctor is designing the therapy part of a particular protocol. When he is specifying the drugs to be used, the program checks for simple constraints such as whether the dosages are within generally accepted bounds. When he finishes specifying the drug schedule, the program, using its knowledge of the expected toxicities, asks for rules for dealing with each of these toxicities and would thus automatically make this section complete and also check whether appropriate study parameters have been specified to monitor the toxicities. This approach of formalizing the structure of the protocol and then leading the doctor through it has the advantage that one doesn't have to wait till the error is made and then detect the error. By a well designed structure such errors (of omission) are avoided in the first place. Such an approach however isn't sufficient. This takes care of only having parts of the protocol filled in. To check them for consistency the program has to apply the constraints it "knows" to the available data. For instance when checking for dose modification rules the protocol structure forces the designer to specify the relevant dose modification rules. To further check the treatment for possible dangerous toxicity the program must use its knowledge of the toxicities caused by the drugs, their time frame and the drugs' schedule and combine them to identify periods when severe toxicity is expected and warn the physician about these periods.

1.6 Brief discussion of problems encountered

A program with an aim of helping the design of the complete protocol is limited by the lack of knowledge in the domain of cancer. This knowledge can be roughly divided into knowledge about drug processes, disease processes and knowledge of the toxic processes. Most of such knowledge is either absent or exists in the form of heuristics. The data that exists is primarily empirical with very few mechanisms of action that are completely understood. The few that are understood are not easy to formalize because they require simulations of complex bio-chemical processes i.e. there is little categorical knowledge, which can be formalized quickly, in a form that is useful. This is exactly the kind of knowledge however that is the basis of the *rationale* of a new therapy. This means that understanding of the rationale of the therapy by programs must, at this stage, be limited to rather rudimentary levels. The scope of this project is restricted to dealing with issues of consistency, ambiguity and completeness of the protocol. However, not being able to deal with the over all intent of the therapy restricts the extent to which the program can check the consistency and the completeness of the specifications because many of the justifications derive from the over all intent of the treatment (e.g. whether the treatment is curative or palliative). Other problems which make even the limited scope of this project harder to achieve are:

Subjectivity of medicine: While it is easy to quantify many of the parameters of measurement (such as blood counts) there are a few important ones that are subjective in nature, notably having to do with measurement of the extent of the disease, and those which describe the location and the spread of disease. Specification of what is "severe", "life-threatening" or "mild" is still subjective in most cases. Such descriptions also sometimes require the use of naive spatial/geometric knowledge (for instance in describing where something is located, whether something is movable or not) and nothing short of a system of common sense knowledge of the anatomy and physiological processes would deal with this problem (of its description and reasoning with it) in a satisfactory way.

Variations within protocols: We have alluded earlier to the difficulty in understanding the rationale or the intent of the therapy. Since many of the justifications of the details of the therapy depend on such a rationale it is hard to formalize what is "proper" and what is not, as is required in judging whether a certain combination of toxicity modification rules is admissible or not, in the interest of the patient. Such rules are general to some extent but are frequently overridden by specific exceptions to the norm. These are then hard to incorporate consistently, in the absence

of a complete understanding of higher level rationale (or model) of therapy in general.

Computational problems: Apart from issues relating to the domain of cancer there are some which are related to the nature of representation and control structures. The need to change, edit, store, interrupt and continue the protocol are only some of the places where they arise. Others include naming objects, representing constraints in a structured way and I/O.

1.7 Lessons learnt

What have we learnt from this project? About the nature of the domain, the state of art of the knowledge of it, the relevance of AI techniques to it, issues relating to the extent and the depth of the of representation and of the distribution of effort in an engineering project of this kind? We just briefly review the issues, most of which are discussed in more detail in the concluding chapter.

The relevance and efficacy of AI techniques to the domain of cancer (more precisely the domain of *design* of cancer therapy) are largely limited by the absence of significant fundamental knowledge of the mechanisms of cancer. field.

Regarding representation: We have used a very simple representation scheme to represent knowledge in this domain (lung cancer) and have found the belief reaffirmed that as long as one is working with a specific domain it is better to fashion one's own representation language, because the domain specific trade offs make it worthwhile not getting committed to a language which maybe more powerful than one needs and may have been designed for a different domain. Another issue regarding representation has to do with the levels of representation and what limitations are imposed by having inadequate levels.

In a project of this kind which has to be designed with users in mind, the user engineering part of the project takes up a significant part of the project resources. In this project roughly one third of the programming effort was devoted to implement a reasonable I/O which makes it easier for users unfamilar with computers to use the program.

2. A demonstration of the system

The following session illustrates a typical session with the doctor. To be able to annotate the session and point out relevant issues and drawbacks, we have restricted this chapter to only excerpts from a complete session, which runs into many pages. The characters in italics are the user responses.

```
Welcome to the protocol writer's assistant....

Do you want the general information?YES

        This program helps express and design a protocol in the domain of
lung cancer. Currently it can deal with with the therapy, eligibility, dose
modifications, study-parameters and the statistical parts of the protocol
design. Once expressed the protocols can be stored, retrieved, displayed and
printed. The program acquires data through questions and answers. In general
a space needs to be typed after the answer to confirm it. Before confirming
the answer, it can be modified using the "delete" key. A "?" will give
options for a given question. For the statistical part "explain" will give
definitions of parameters. At each stage typing "abort" or "quit" will
return to the top level. It is not always possible to continue after
aborting and the program will warn the user if such is the case. New
protocols are not saved automatically. Before quitting the user should store
any protocols to be saved through the "store" command. Protocols can be
continued in the next session by reading them in and "continuing" them. Of
course any number of earlier protocols can always be displayed.

Type a space to continue......
```

After the initial message the program begins to instantiate a protocol under user control.

```
What do you want to do? CREATE-NEW-PROTOCOL

What is the name of the protocol? : Comparing Intensive CCM with standard CCM
What part of the protocol do you want to instantiate? THERAPY

Now instantiating the THERAPY section of the protocol....

This is the therapy part of the protocol.....
Are the patients required to be stratified? YES
What is the criteria for stratification? AGE

   category 1 :  AGE  < 50
   category 2 :  AGE  > 50

Are there any more criteria for stratification? YES

What is the criteria for stratification? KARNOFSKY-STATUS

  category 1 :  KARNOFSKY-STATUS   < 40
  category 2 :  KARNOFSKY-STATUS  50 TO 70
  category 3 :  KARNOFSKY-STATUS   > 70
```

While defining these categories the program ensures that they are defined properly (e.g. that they do not overlap, cover the whole range) If in the above definition the user had tried to quit after defining the second category the program would warn him that the categories aren't logically complete.

```
Are there any more criteria for stratification? NO
Are the patients required to be randomized at this stage? YES
Enter the name of the regimen to which the patients are assigned after
randomization :Regimen A

Are there any more regimens? YES
Enter the name of the regimen to which the patients are assigned after
randomization :Regimen B
Are there any more regimens? NO

Now looking at REGIMEN A  ........
```

Do you want to specify the name of the treatment? *YES*
Enter the name of the treatment and terminate it with a <CR> : *CCM Intensive*

What is the type of treatment? *CHEMOTHERAPY*
Do you want to specify the details of the treatment now? *YES*
Do you want to specify the cycle-length of the treatment? *YES*
Enter the length of the treatment cycle : *6 WEEKS*

Enter the details of the treatment in the format:

Drug-name drug-dosage unit administration-mode <optional frequency> schedule
CYCLOPHOSPHAMIDE 1500 MG/SQ-METER

1500.0 MG/SQ-METER is greater than the upper bound of
CYCLOPHOSPHAMIDE DOSAGE specified as 1200.0 MG/SQ-METER

Type any character to continue

The program checks the dosage for being within the bounds generally accepted. Here it is checking just the single dose limit but it has more general knowledge of the time frame in which the drug can be given and it then checks for the time dependent limits on the dose.

CYCLOPHOSPHAMIDE 1200 MG/SQ-METER IV DAYS 1 AND 22
Are there any more drugs? *YES*
CCNU 70 MG/SQ-METER PO DAY 1
Are there any more drugs? *YES*
METHOTREXATE 15 MG/SQ-METER PO BIW WEEKS 2 3 5 AND 6
Are there any more drugs? *NO*
During the treatment do you want to evaluate the patient after a specific period

After the drug schedule is specified the program goes on to ask questions regarding the general plan of the treatment. In general the details of the drug schedule (as given above) can also be specified later. In the part of the session that follows the program tries to generate a complete specification of what is to be done at each branch point of the treatment, depending on the possible responses the patient may have at that branch point. The abbreviations PR, CR, NC, PD in the following lines mean partial response, complete response, no change and progressive disease respectively.

How long is the treatment to continue before an evaluation is made? *7 CYCLES*

After 7.0 CYCLES what is the type of the test to be performed

to evaluate the patient's condition? *EVALUATE-PATIENT-CONDITION*
Now looking at CR .....

If there is CR then *CHANGE-TREATMENT*
What is the name of the next treatment to be administered? *avp*
Now looking at PR .....
Note that the treatment avp is not defined yet.

If there is PR then *CHANGE-TREATMENT*
What is the name of the next treatment to be administered? *avp*
Now looking at NC .....

If there is NC then *OFF-STUDY*
Now looking at PD .....

If there is PD then *OFF-STUDY*

The program is now completing the branch which is generated by changing the treatment, by asking questions which specify the treatment and later more questions will determine the possible duration and outcome after this treatment.

Now looking at the changed treatment AVP.....

What is the type of treatment? *CHEMOTHERAPY*
Do you want to specify the details of the treatment now? *YES*
Do you want to specify the cycle-length of the treatment? *YES*

E nter the length of the treatment cycle : *6 WEEKS*

Enter the details of the treatment in the format:

Drug-name drug-dosage unit administration-mode <optional frequency> schedule
ADRIAMYCIN 60 MG/SQ-METER IV DAYS 1 AND 22

Are there any more drugs? *YES*

And so on ..... The specification will end when all branch points will have been follwed to their ends, and when at each branch point all the possible events have corresponding actions. After specifying the treatment the program can display the treatment plan as a schema as well as in the typical written style of a protocol.

Which part of the protocol do you want displayed? *TREATMENT*

6.0 DRUG TREATMENTS

6.12 Regimen A

Treatment CCM Intensive

Cyclophosphamide 1200.0 Mg/sq-meter (max dose 1200 Mg/sq-meter) IV Days 1, 22

Ccnu 70 Mg/sq-meter PO Days 1

Methotrexate 15 Mg/sq-meter (max dose 30 Mg/sq-meter) PO Biw weeks 2,3, 5 and

6.121 After 8 cycles those patients who experience complete response or partial response will change treatment to AVP. Those who experience no change or progressive disease will go off study.

Treatment AVP

Adriamycin 60 Mg/sq-meter (max dose 110 Mg/sq-meter) IV Days 1, 22

Vincristine 1 Mg/sq-meter (max dose 2 Mg/sq-meter) IV Days 1, 22

Procarbazine 100 Mg/sq-meter (max dose 400 Mg/sq-meter) PO Days 1- 10, 22 -

6.122 Those patients who experience complete response , partial response or no change will continue on treatment. Those who experience progressive disease will go off study.

6.13 Regimen B

Treatment CCM Standard

Cyclophosphamide 700 Mg/sq-meter (max dose 1200 Mg/sq-meter) IV Days 1

Ccnu 70 Mg/sq-meter PO Days 1

Methotrexate 15 Mg/sq-meter (max dose 30 Mg/sq-meter) PO Biw weeks 2,3,5 and

6.131 After 7 cycles those patients who experience complete response or partial response will change treatment to CCM. Those who experience no change or progressive disease will go off study.

Treatment CCM

6.132 Those patients who experience complete response , partial response or no change will continue on treatment. Those who experience progressive disease will go off study.

Type a space to continue...

Next we go on to specify the eligibility part of the protocol.

Now instantiating the ELIGB section of the protocol....

This is the eligibility section of the protocol
what is the class of the disease? *CARCINOMA*
What is the site of the tumor? *LUNG*
What is the type of carcinoma? *ADENO*
Does the diagnosis need to be confirmed histologically? *YES*
Is the sputum cytology required? *NO*
Is the tumor to be staged? *YES*
What is the stage of the tumor? *II*
Do you want to specify any more primary tumors? *NO*
Do you want to specify the presence of abnormal lymph-nodes ? *YES*
Are the nodes to be clinically-staged? *NO*
Are there any specific sites for the the metastasis? *YES*
Enter the site : *BRAIN*
Are there any more sites? *NO*
Is the disease required to be measurable or evaluable? *YES*
Do you want to specify the disease-activity? *NO*
Do you want to specify conditions regarding prior therapy? *YES*
What kind of prior-therapy do you want to rule out? *CHEMOTHERAPY*
Do you want to specify a time limit earlier than which the patient may have undergone the above therapy? *YES*

Enter the time limit in months: *12 MONTHS*

Are there any exceptions to the general class of
therapy that you have specified? *NO*
Do you want to rule out any other kind of prior therapy? *NO*
Do you want to specify the range of the age of the patient? *YES*
Do you want to specify the upper limit for the age? *YES*

Enter the upper limit : *70 YEARS*
Do you want to specify the lower limit for the age? *YES*

Enter the lower limit : *40 YEARS*
Do you want to specify the sex of the patient? *NO*
Do you want to specify the general performance of the patient by a
performance status? *YES*
Do you want to specify the Karnofsky status? *YES*
Do you want to specify a lower limit? *YES*

Enter the lower limit : *50*

Do you want to specify an upper limit? *NO*

Do you want to want to rule out any other disease/dysfunction? *NO*

The drug ADRIAMYCIN used in treatment AVP causes
CARDIAC-DYSFUNCTION. Do you want to rule out CARDIAC-DYSFUNCTION? *YES*
The program triers to maintain consistency between different parts of the clinical trial design. For instance as above it tries to assure that all expected toxicities are screened out at the time of eligibiltiy. After the specification is completed the program can display the section as in a typical protocol.

Which part of the protocol do you want displayed? *ELIGB*

3.0 SELECTION OF PATIENTS

3.10 Patients with stage II adeno carcinoma, with possible metastasis in the brain will be eleigible for this protocol.
3.11 Patients should have measurable or evaluable disease.
3.12 Patients must be older than 40.0 years and younger than 70.0 years.

3.13 Patients must have a karnofsky status greater than or equal to 50.0.
3.14 Patients with chemotherapy in the last 12 months are ineligible.
3.15 Patients with cardiac-dysfunction are ineligible.
3.16 Patients must live within 100.0 miles of the treatment centre.
3.17 Informed consent from patients is mandatory.

Type a space to continue...
Next we see the specification of dose-modification rules....

Do you want to specify the rules for dealing with various toxicities now? *YES*

The following session sets up rules for handling various toxicities

CARDIAC-TOXICITY may occur in REGIMEN A
and is caused by ADRIAMYCIN
It's seriousness can be MILD, MODERATE OR SEVERE

What actions do you want to be taken
to handle CARDIAC-TOXICITY? *REDUCE-DRUG-DOSAGE*
In the following rules for dose modification, does the
reduction apply to the last dose given or to
the base-line-dose? *BASE-LINE-DOSE*
This is just to ensure clarity in case the reduction is applied more than once, so that the reduction in this case is always computed from the initial (base line) dose.

Do you want to see the definitions of the different severity levels? *YES*

CARDIAC-TOXICITY

MILD : Intermittent arrythmia

MODERATE : Change in conduction

SEVERE : Congestive heart failure or persistent arrythmia

Do you want to redefine the severity levels? *NO*
In general when degrees of toxicities are defined using text (as in most symptomatic toxicites) as

above, the program has no way to check them for consistency as it doesn't have the ability to understand natural language. However the toxicites monitored by a parameter lend themselves more easily to checking.

```
CARDIAC-TOXICITY        ADM
-----------------------------
MILD                    100
MODERATE                 50
SEVERE                    0

ALOPECIA may occur in REGIMEN 1
 and is caused by ADRIAMYCIN
It's seriousness can be MILD, MODERATE OR SEVERE

What actions do you want to be taken
to handle ALOPECIA?dDoO-NOTHING  The level of seriousness for ALOPECIA can be seve
Do you really wish to do nothing? YES

HEMATOLOGIC-TOXICITY may occur in REGIMEN 1
 and is caused by ADRIAMYCIN, CYCLOPHOSPHAMIDE, AND PROCARBAZINE
It's seriousness can be SEVERE, MODERATE OR MILD

What actions do you want to be taken
to handle HEMATOLOGIC-TOXICITY? REDUCE-DRUG-DOSAGE

In the following rules for dose modification, does the
reduction apply to the last dose given or to
the base-line-dose? BASE-LINE-DOSE

HEMATOLOGIC TOXICITY

   WBC-COUNT (/CUBIC-MM)        ADM
   -------------------------------
       > 4000                   100
      3999 to 2500               75
      < 2500              50
```

And so on ..... In the actual protocol dose modifications for drugs other than ADM will also be specified. After the specification is complete it can be displayed in the typical format.

Which part of the protocol do you want displayed? *DOSE-MODIFICATIONS*

7.0 Dose Modifications

7.11 CARDIAC-TOXICITY (Regimen 1)

The following severity levels of the above toxicity are expected:

MILD : Intermittent arrythmia

MODERATE : Change in conduction

SEVERE : Congestive heart failure or persistent arrythmia

The following dose reductions will be effective for the different levels of severity:

CARDIAC-TOXICITY	ADM
MILD	100
MODERATE	50
SEVERE	0

7.12 HEMATOLOGIC-TOXICITY (Regimen 1)

The following dose reductions will be effective for the different levels of severity:

WBC-COUNT (/CUBIC-MM)	ADM
> 4.0E+3	100
3.999E+3 to 2.5E+3	75
< 2.5E+30	50

7.13 HEPATIC-DYSFUNCTION (Regimen 1)

The following dose reductions will be effective for the different levels of severity:

BILIRUBINT (MG%)	ADM
< 1.5	100
1.6 to 3.0	75
> 3.0	0

7.15 MUCOSITIS (Regimen 1)

No action will be taken for the toxicity.

Next we see the use of the statistical advisor. The particular statistical terms used are defined and explained in the statistical section .

What do you want to do? *COMPUTE-STATISTICAL-PARAMETERS*

This the statistical advisor.....

SET-UP-DEFAULT-VALUES? *YES*

What do you want to do? *COMPUTE-PARAMETER*

What is the parameter that you want to look at? *CHANCE-OF-DETECTION*

Parameter of interest : CHANCE-OF-DETECTION :

The program first shows all the parameters needed to compute the specified one.

Parameters needed :-

NUMBER-OF-REGIMENS : 2.0
NUMBER-OF-STRATA : 3.0
RATIO-OF-EFFICACY : 1.5
HALF-LIFE : 2.0 4.0 6.0 YEARS
HAZARD-RATE : 0.347 0.173 0.116 /YEAR
RATIO-OF-ALLOCATION : 0.5
PROPORTION-ALLOCATED : 0.4 0.4 0.2
ACCRUAL-RATE :
P-VALUE : 0.05
ACCRUAL-PERIOD : 2.0 YEARS
FOLLOW-UP-PERIOD : 2.0 YEARS

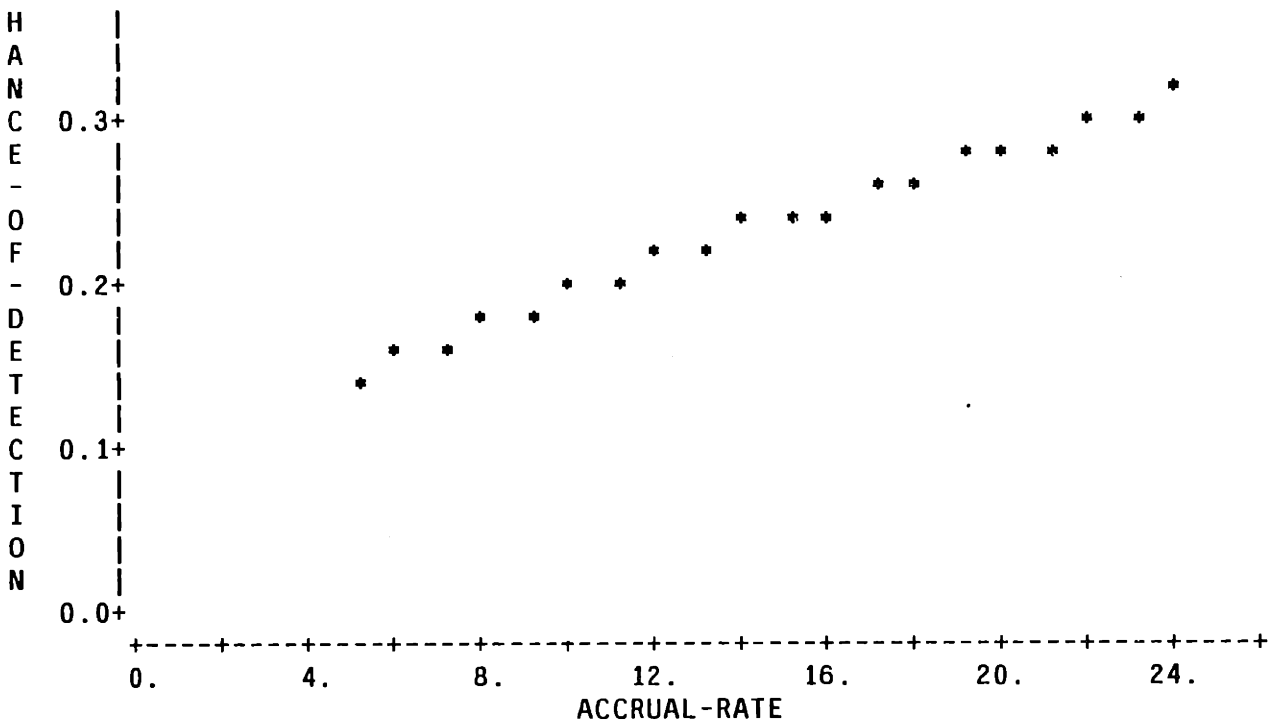
Most of the values of the parameters have been set by default for this session, but in general the program acquires them through queries

```
Is the value of ACCRUAL-RATE known? YES

Enter the value of the accural rate : 10 PATIENTS/MONTH

Parameter of interest : CHANCE-OF-DETECTION : 0.172611

What do you want to do now? RECOMPUTE-WITH-CHANGES

Please change whatever parameters you wish and type "finish"
when you are done.

> ACCRUAL-RATE : 20 PATIENTS/MONTH
>  FINISH


CHANCE-OF-DETECTION : 0.240593
```

We see above the recomputed value of the chance of detection which has increased because the patient accrual rate has increased. To see the effect of a whole range of change we can plot parameters.

```
What do you want to do now? PLOT-PARAMETERS

 CHANCE-OF-DETECTION  VS.  ACCRUAL-RATE

Do you want to specify a range for the ACCRUAL-RATE? YES

 Enter the lower limit for the ACCRUAL-RATE : 5 PATIENTS/MONTH
 Enter the upper limit for the ACCRUAL-RATE : 25 PATIENTS/MONTH
Computing data for the plot.......

ACCRUAL-RATE          CHANCE-OF-DETECTION

       11.0                   0.204
       12.0                   0.213
       13.0                   0.222
       14.0                   0.231
       15.0                   0.24
       16.0                   0.248
       17.0                   0.257
       18.0                   0.265
```

19.0	0.273
20.0	0.281
21.0	0.289
22.0	0.297
23.0	0.305
24.0	0.313



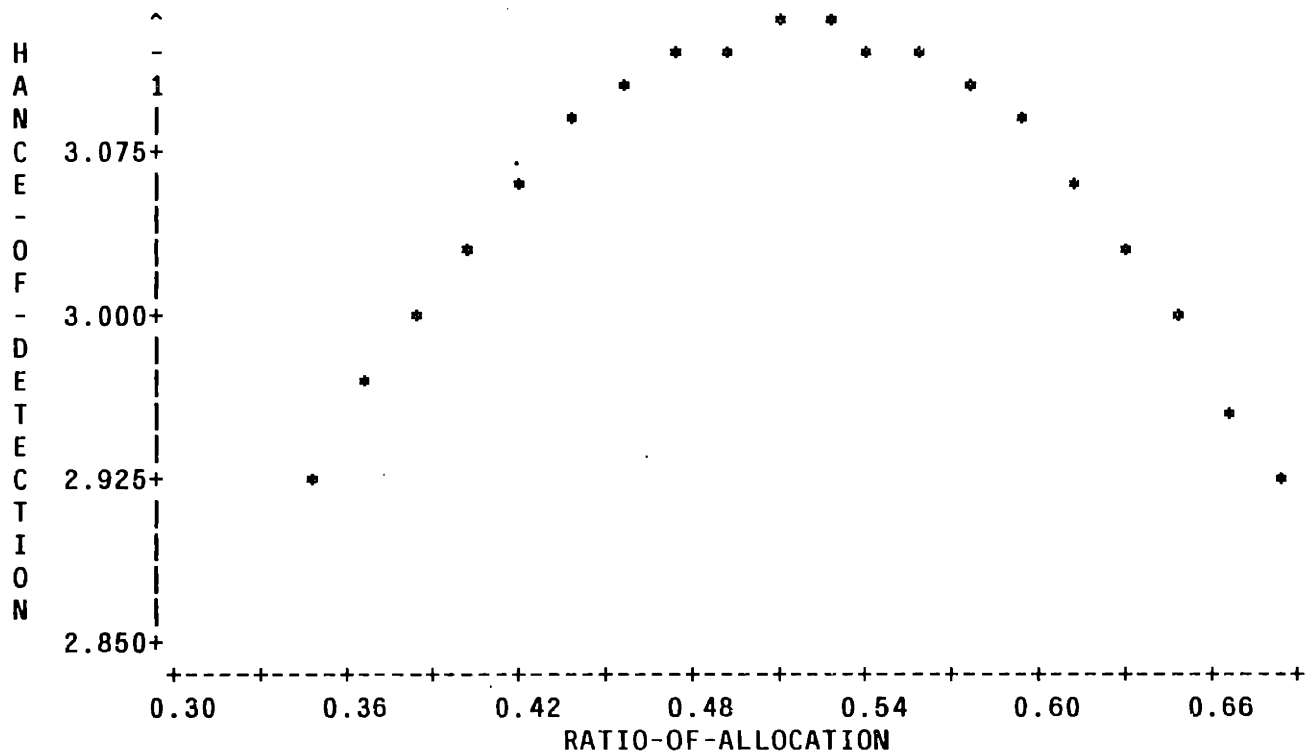
Do you want to save the plot? *YES*
Saved as plot 1

We see the expected result that the chance of detection increses with incresing patients. Similarly we can see th effect of allocating differing protoportions of patients to the different treatment arms and we see the intuitive result that an equal allocation is the best in this case.

Computing data for the plot.....

RATIO-OF-ALLOCATION	CHANCE-OF-DETECTION
0.35	0.292
0.368	0.297
0.385	0.3
0.403	0.304
0.42	0.306
0.438	0.309
0.455	0.31
0.473	0.312
0.49	0.313
0.508	0.313
0.525	0.313
0.543	0.312

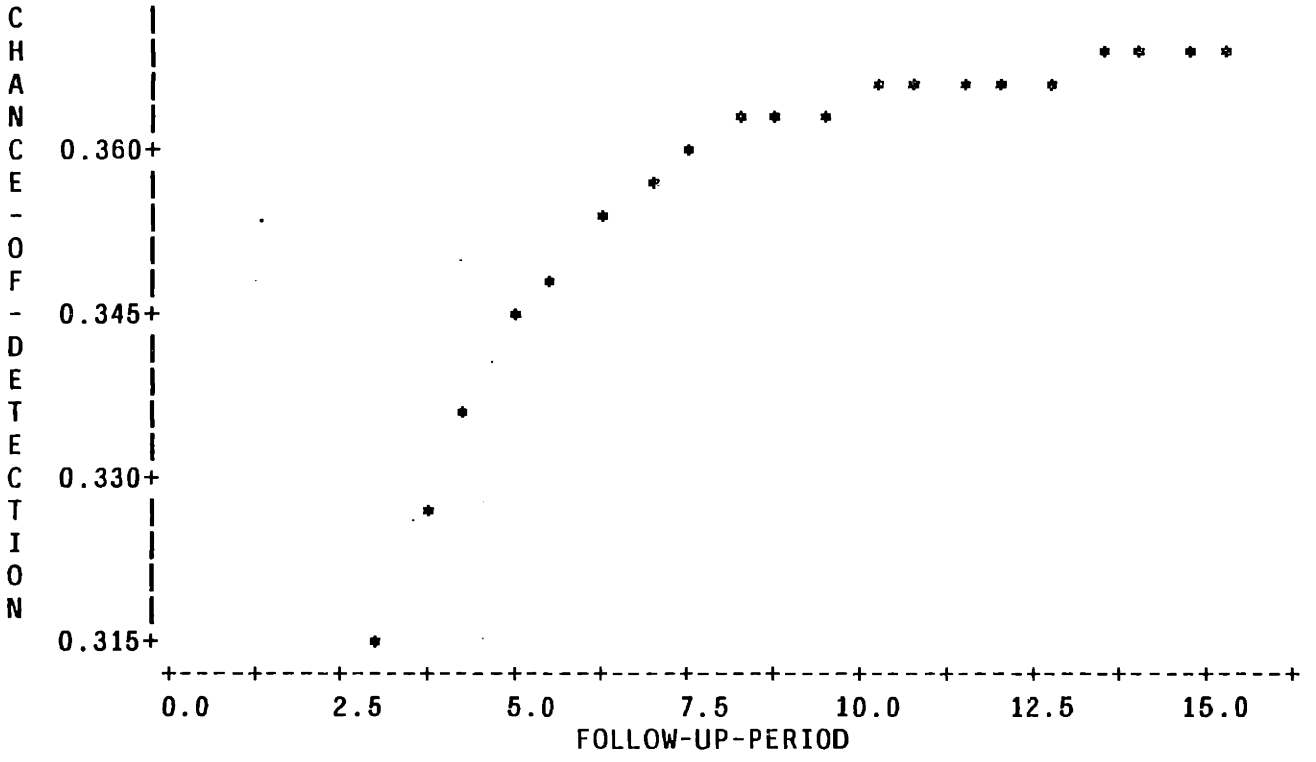
0.56	0.312
0.578	0.31
0.595	0.308
0.613	0.306
0.63	0.303
0.648	0.3
0.665	0.296
0.683	0.292C



Finally we see the effect of increasing the follow-up period on the chance of detection. As expected the chance of detection levels off as most of the patients are dead within a few half live of the disease.

Computing data for the plot.....FOLLOW-UP-PERIOD	CHANCE-OF-DETECTION
3.0	0.316
3.65	0.328
4.3	0.337
4.95	0.344
5.6	0.349
6.25	0.353
6.9	0.357
7.55	0.359
8.2	0.362
8.85	0.363
9.5	0.364
10.1	0.365
10.8	0.366

11.4	0.367
12.1	0.367
12.7	0.367
13.4	0.368
14.0	0.368
14.7	0.368
15.3	0.368

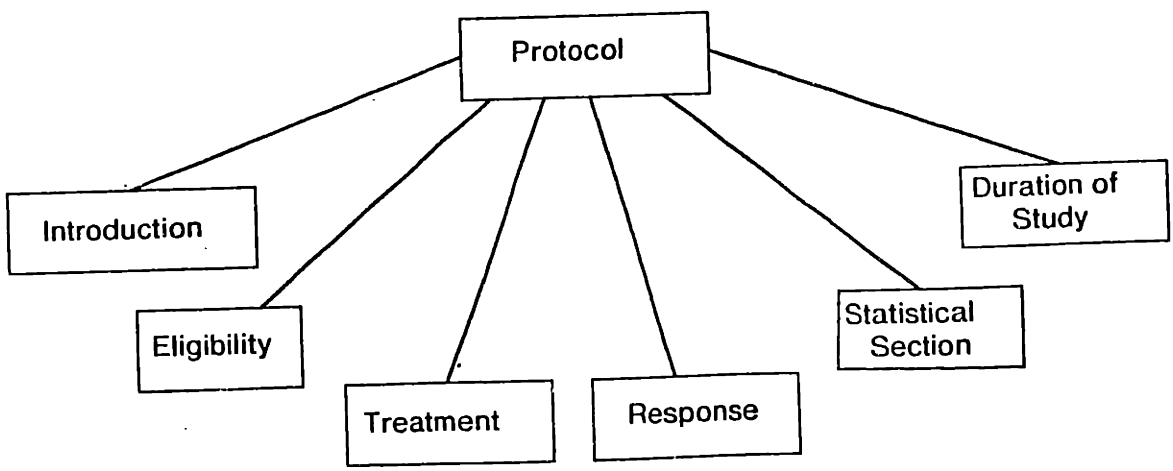


3. The Protocol

This chapter describes the structure of the protocol and the dependencies between different parts of it. As we describe the protocol we also note some of the related problematic issues, which are discussed later (chap6) in detail. We are dealing with protocols relating to chemotherapy only. We describe phase 3 protocols (which compare different treatments) because these are usually the most complex.

The protocols typically contain an introduction explaining the motivation and the philosophy behind the therapy, an eligibility section which spells out the requirements for patients to be eligible for the study, a therapy section detailing the treatment (the most important part of the protocol), a section specifying the study parameters, a section specifying the criteria for response, a section specifying the rules for the duration of treatment and a statistical section justifying the logical and statistical design of the protocol. In addition to these sections there are other sections relating to some administrative procedures. We now proceed to examine each section in more detail. The discussion of the statistical section comprises the next chapter.

Fig. 1. The Protocol



3.1 Objectives of study

The objectives spell out what hypotheses are to be tested and give some idea of the general related observations that are to be made.

3.2 Introduction and Scientific background

This section explains why the particular therapeutic experiment is being carried out. This contains relatively technical justifications of the current treatment in the light of previous results. This is of importance to the designer and his peers in that it determines the framework in which to evaluate any published results. Although it contains the high level rationale of the treatment, it is frequently ignored in the use of the protocol as a working document.

3.3 Eligibility

This section on selection of patients defines the patient population to be studied. It contains selection criteria which define the type, extent of disease expected, the type of prior therapy allowed, age restrictions, requirements for pre-treatment physiologic functions (e.g. renal, liver, cardiac, bone marrow status), requirements for presence of measurable lesions and the universal requirement for informed consent as a precondition to entry into the study. In addition to these general requirements there may be requirements for minimal life expectancy and other criteria specific to some protocols. Some examples of typical rules for eligibility are:

1. Only patients with a histologically confirmed small cell carcinoma of the lung will be admitted to the study.

2. Patients with age > 71 or prior chemotherapy or radiation will be excluded.

3. Patients must have adequate renal and bone marrow function.

4. Patients must live within 100 miles of the hospital.

5. Patients must have measurable or evaluable disease.

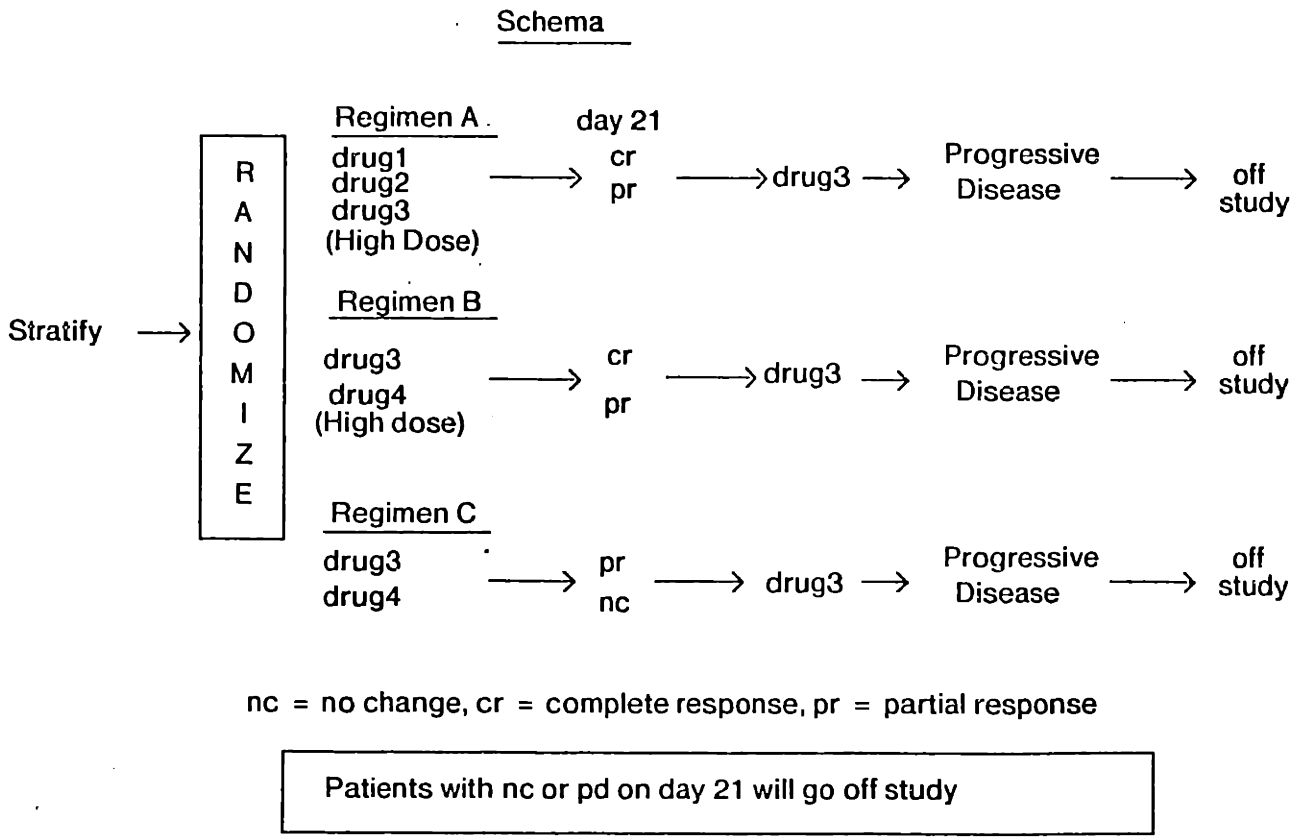
It is clear that some of the rules which are disease specific derive from the intent and the philosophy of the protocol. Others which are are patient specific derive from considerations of

toxicity in the treatment and still others derive from the need to have a uniform group so that prior therapy etc. will not interfere with a statistical evaluation of the result. Apart from these categories there are some other rules (such as requiring informed consent from the patient) which are just mandatory and will not be discussed here. Note that in the above rules it is not clear what "adequate" renal and liver function specifies and how long ago "prior" is.

3.4 Therapy

The section on the treatment plan is the one most frequently referred to by physicians caring for a patient. It contains details of the schedule and dose of the chemotherapeutic agents with appropriate provisions for increasing or decreasing the dose depending on observed toxicity.

Fig. 2. Schema of Typical Treatment Plan



The therapy involves randomization into two or more treatment groups which receive different treatments so that the comparative effects of the treatments can be observed. A *schema* of a simple treatment plan can be seen in the figure below. In regimen A the patients receive drug1, drug2 and drug3 according to the given schedule and if there is a partial or complete response by day 21 they go on to drug3 for follow-up treatment; otherwise they go off-study . In the follow up treatment the patients go off the study if they develop progressive disease again. The explanations of regimen B and C are similar. At this level of specification the program can check for planning errors. For instance, in regimen C it is not clear what happens if patients have a complete response by day 21. Notice that this protocol is designed for the treatment of a disease which is lethal ultimately because everyone is expected to encounter progressive disease at some stage and is followed until death to measure the duration of survival.

A more detailed specification of the treatment includes the schedule and procedures of administration of the drug and instructions to modify dosage of drugs in case of various toxicities. For example:

CCM:(Initial treatment)

Cyclophosphomide 700 mg/sq-meter IV days 1 and 22
CCNU 70 mg/sq-meter PO day 1
Methotrexate 15 mg/sq-meter PO biw wks. 2,3,5 and 6

AVP(Follow-up treatment)

Adriamycin 60 mg/sq-meter IV day 1,22
Vincristine 1.4 mg/sq-meter IV day 1, 22
Procarbazine 100 mg/sq-meter PO days 1-10, 22-30

Dose Modifications for Renal Dysfunction (Initial and follow-up treatment)

	% of full dose to be given	
Serum Creatinine (mg%)	Methotrexate(%)	Cyclophosphamide(%)
< 1.5	100	100
1.5 - 2.5	100	50
2.5 - 3.5	100	0
> 3.5	50	0

This is the most complex part of the protocol. At the simplest level of checking for mistakes the program has to make sure whether the drug dosage is within bounds, whether the drug schedule is consistent with the treatment cycle schedule etc. At a more difficult level it has to see whether there are instructions for dealing with expected toxicities. This involves checking for or setting up instructions which deal with the detection and palliation of toxicities at their various levels of severity. The program tries to assure that there is an instruction for all reasonable contingencies which are expected from a knowledge of the drugs in use and their side-effects and keeps track of the interactions among different dose modification rules.

To make the protocol consistent as a whole the program has to see whether the therapy section communicates appropriately with other sections. For instance the fact that a certain toxicity is expected due to a certain drug implies that there must be a specification of a study parameter to monitor the toxicity at adequate intervals and the eligibility section must screen patients who may be at a severe risk due to poor health.

3.5 Study Parameters

This section is fairly simple and just contains a schedule of parameters to be monitored. This section mainly derives from the therapy section and it has to be complete in the sense that for each expected toxicity in the therapy section it must have a corresponding parameter to monitor it. Apart from parameters to measure toxicity, this section also specifies parameters relevant to the intent of the study.

3.6 Measurement of Response

This section defines various terms used to measure patient response. Typical definitions are:

1. Complete response: The disappearance of all signs and symptoms of the disease for at least 4 weeks.

2. Partial Response: More than 50% reduction in the sum of the tumor areas (determined by multiplying the two greatest perpendicular diameters of all measurable lesions) without an increase in the size of any mass or the appearance of new lesions; or more than 75% reduction in the estimated areas of poorly measurable lesions.

3. Progressive disease: More than 50% increase in tumor size measured in a fashion described above.

4. No response: if none of the above.

This section illustrates the problem involved with variations in descriptions. The above definitions can be described in a number of ways and the best that can be done at this level is to provide a library or a template of definitions from which the doctor can then choose the ones he likes. The definitions will not be "understood" by the program in as much as they are just strings of characters as far as the program is concerned.

3.7 Duration of study

This section spells out the endpoints of interest and expresses the rules for the duration of the study in terms of these endpoints. It should be noted that the duration of the study can exceed the duration of treatment (for purposes of measuring response). Rules about the duration are implicit in the *schema* of the treatment which contains rules about what is to be done at different stages of the treatment depending on the patient response. A typical rule is:

> Patients who achieve a partial response should continue to receive chemotherapy as stipulated for an indefinite period until disease progression is noted at which point they will be taken off study.

It should be noted that this rule implicitly contains information about the nature of the particular disease process (in this case fatal), in the sense that it is not considered necessary to account for the possibility of complete response.

3.8 Summary

This chapter describes the structure of the protocol and the dependencies between different parts of it. The different parts are: The aims and objective, the therapy section, the eligibility section, the study parameter section, the measurement of the effect section, the section defining the duration of study and the section dealing with the statistical design of the clinical trial.

While the aims and the objective define the higher level rationale, which is necessary for an overall model of the *intent* of the protocol, these are hard to formalize. The therapy section describes the treatment in detail (the schedule of drugs) and most importantly the dose modification rules for the toxicities which are expected. The eligibility part has rules to screen patients for entry into the clinical trial. The rules derive partly from the specific treatment because patients have to be deemed fit enough to undergo the different expected toxicities and partly from the intent of the study which defines the patient population to be studied. The study parameters and the measurement of effect section together determine what is to be observed and what constitutes a valid measurement. The data from the study parameters helps both to regulate the treatment (through the dose modification rules) and to evaluate the treatment for study purposes later. The section specifying the duration of study determines how long the patients are going to be treated and studied. The next chapter describes the statistical section which ensures that the expected number of patients to be accrued is adequate to detect expected differences in the treatment with a degree of confidence generally accepted in statistical circles.

The difficulties relating to the formalization of the protocol include the lack of an over-all model of the intent of the treatment, and the subjective descriptions still in use to characterize the extent of disease and the measurement of effect.

4. The statistical design of the protocol

4.1 Basic issues

The central statistical question in designing a clinical trial is "Will there be enough patients to discriminate among the treatments?" A related question is "Given the expected difference in the treatments and the expected number of patients to be accrued in the study, what is the chance of detecting the difference?" If the number of patients is not adequate then the observed differences in the therapeutic effect may be due to chance factors or an actual difference may be obscured by chance variations in the population of patients in the two groups. Hence the need for a good statistical design.

The answer to the above question usually involves a statistical analysis of certain parameters relevant to the treatments, e.g., how different are the treatments (as estimated)? How long is the course of the disease? What is the number of expected deaths? What is the degree of confidence required? and many similar questions.

If all the relevant parameters were known with accuracy, a single calculation would be enough to suggest the size of the population. But due to the very nature of the trial, certain parameters (such as how different are the studies?) can only be estimated. The purpose of the study *is* to find this parameter. Therefore the design process usually involves coming up with a number of different designs and then evaluating them statistically using the estimates of various parameters. Due to the complexity of the calculations and the spread of the estimates, this can be a laborious process if done by hand (with a calculator). The designer of a study needs to know the answer to questions like the above and typically not being a statistician himself, he will consult a statistician.

We have attempted in this statistical part of the program to bring that statistical expertise directly to the doctor. With the facility than we provide the doctor can play around with different parameters and look at the results. For instance he can explore situations such as : What if the study goes on for two more years? If the treatments are only half as different as he thinks, how would that affect the sensitivity of the study. Apart from providing specific calculations of the above sort the system can plot graphs between parameters. and This enables the user to see the effect on a whole range, giving him a kind of intuition which is hard to capture with point calculations.

We now go deeper into the basic theory so that we can formulate the statistical problem much more precisely and look at the solution that we have implemented, its power and shortcomings. We will introduce just enough statistical terminology so that we can describe the issues and discuss them in the domain of medicine. References are provided for the more interested reader.

4.2 Approach to the statistical problem

The problem of the clinical trial of two different therapies is, in the language of statistics, the problem of "Hypothesis testing" namely whether the treatments are same or different, referred to as the *null hypothesis* and the *alternate hypothesis* respectively. Considerable literature exists about the statistical problem of hypothesis testing, along with various simplified solutions for abstract domains. However interpreting their assumptions in the domain of medicine in general and cancer therapy in particular is a difficult problem, not to speak of the problem of estimating some of the parameters on which the result crucially depends. We describe the simplified theory of hypothesis testing first, followed by the problems of estimating the parameters.

4.3 Hypothesis testing

Let us assume that we are comparing two actual treatments and the two hypotheses about them are

The null hypothesis H0: The treatments are the same.

The alternate hypothesis H1 : The treatments are different.

Let us call the two treatments as the control and the experimental treatments and assume that we are studying death as the endpoint and that the better treatment will have fewer number of deaths. We will assume further that the experimental treatment is either the same as the control or will reduce the number of deaths, but will not be worse than the control. Such an assumption is made here only to simplify matters and we will deal with the more general assumption later. Usually these assumptions are based on small pilot studies and are not without a basis.

Suppose now that we conduct a clinical trial and actually observe the number of deaths in the two treatments. If the two treatments are similarly effective then we expect roughly the same number of deaths in them and if the new treatment is different then there will be fewer deaths. Let us define delta as the ratio of the deaths in the control treatment to the experimental one and restate our hypotheses as

H0: delta = 1.0

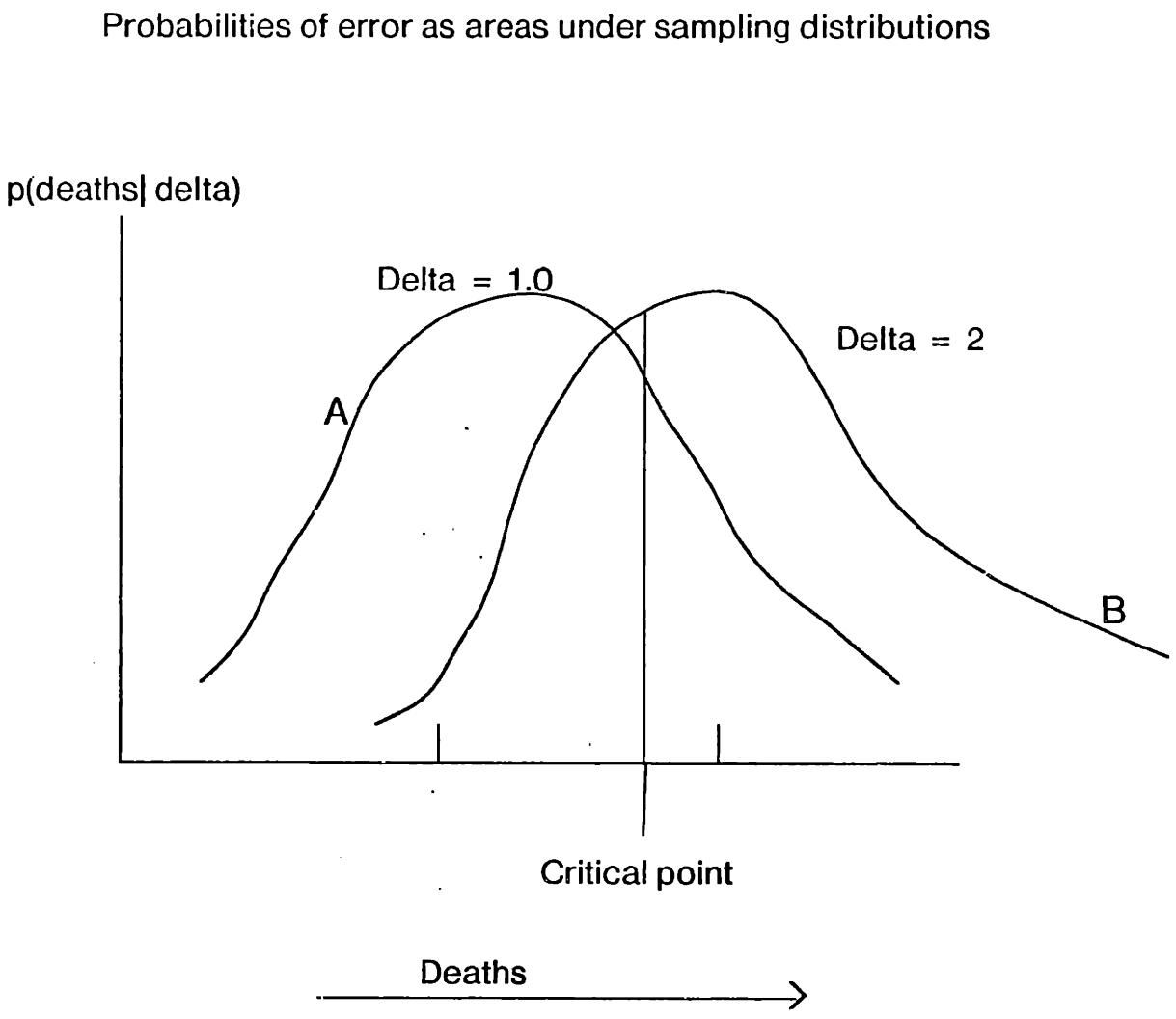
H1: delta > 1.0

In general when we conduct a clinical trial the observed number of deaths will vary according to some distribution about a *mean*. The larger our sample the less the spread. (As governed by the theory of sample means and their variance). Correspondingly delta will also have a mean and a spread about it. Therefore if the treatments are effective to the same degree, then we will have a curve A, with observed values of delta spread around 1.0, and if they are different with a delta of 2.0 (say) then we will have curve B centered around delta = 2.0. The question is the following: Since we conduct the trial only once and observe only *one* value of delta, how do we know which distribution it belongs to? For example if there is a certain observed difference in the treatments (delta > 1.0) then what is the chance that this difference is caused by chance while the treatments are actually similarly effective?

The answer is: We never know for sure, but can give confidence limits regarding the possibility of the observed delta being from a specific distribution. In general the more different the treatments, the fewer patients we need to achieve the same level of confidence and vice versa. Intuitively this is obvious because the closer the treatments are in their effect, the more likely it is that we will confuse just random statistical differences with a real differences and hence need larger sample sizes to iron out the chance fluctuations.

The solution is to choose some point on the delta axis (called the *critical value*) and stipulate that if delta is smaller than that critical value then we take the treatments to be the same and ascribe the variation to chance, if delta is greater than that value we take the treatments to be different because we feel that although the variation could be due to chance alone, it is too large to ignore the possibility of a real difference in the treatments. It is clear that there are many issues involved in choosing the critical point. When we choose *any* critical point and base our judgment

Fig. 3. The error of type 1 and type 2



on it we can make two kinds of errors:

> Error of type 1. alpha (false negative) When the treatments are actually different we can take them to be the same, i.e assume H0 to be true when H1 is true. This is the area to the right of the critical point under curve A.
>
> Error of type 2: beta (false positive) When the two treatments are really the same, we take them to be different i.e. Assume H1 to be true when H0 is true. This is the area to the left of the critical point under curve B.

The probability of the error of type 2 is also known as the *p-value*. Another term of importance is the *power of a test*. It is the probability that if the treatments *are* different then they will be *observed* as such. This is the same as (1 - p-value) In terms of the figure, if the second distribution is valid, then it is the probability that the observed point is on the right side of the critical point. It is usually expressed as a percentage.

How do we choose the critical point so that we minimize the two errors? Firstly, it is clear that minimizing one error maximizes the other and vice versa so that we have to regard one error as more important than the other, reduce its probability to some acceptable value and then compute the probability of the other error. The only way to reduce the probability of the other error is to increase the sample size (thereby reducing the variance).

To make the discussion less abstract, consider that we are testing whether a new medicine is safe or not. If we take the null hypothesis as "The medicine is safe" then the error of type 1 is the error that we think the medicine to be safe when it is not. The error of type 2 is the error that we think it is unsafe while it is safe. Clearly medical ethics dictate that the type 1 error (the p-value) be brought down to some acceptable level so that the probability of an unsafe medicine reaching the market is very low. This is considered more important than having the ability to recognize the medicine as safe. It is for this reason that the statistician usually chooses a critical point so that the the probability of the error of type 1 is less than a certain value (usually 0.05 or 0.01) and then minimizes the error of type 2. By looking at the figure it is clear that once one knows the distribution it is always possible to choose a critical point so that the type 1 error is less than a certain value. In simpler terms it means that one slides the critical point to the right till the p-value is less than a certain value. However once we have chosen this critical point for a particular probability of the type 1 error, the probability of the error of type 2 is determined

automatically and the only way to hold the probability of the error of type 1 constant and decrease the the probability of the error of Type 2 is by increasing the sample size (and decreasing the variance).

The typical (simplified) design proceeds in the following way

1. Determine the distributions of the two deltas. (The value of the delta comes from the guess of the doctor, the variance is computed from the sample size, the larger the sample, the less the variance)

2. Choose the critical point so that the p-value (the value of the type 1 error) is less than 0.05.

3.Compute the probability of the error of type 2 .

If the error of type 2 is greater than (say 20%) then increase the sample size, determine the new (decreased) variance and go back to step 2. otherwise the design is finished.

4.4 Parameter estimation

The above theory and the design process seem simple enough. Where then is the problem? The *problem* is that we have conveniently assumed in the above discussion that we can compute the two distributions (in particular their. variances). This means a) postulating the *nature* of the distribution b) Estimating the variance based on the nature of the distribution. This is where most (if not all) of the issues of interpreting hypothesis testing and estimating parameters are concentrated.

One way to go about it as discussed by Schoenfeld in [Schoenfeld] is to assume that the process of the trial is a Bernoulli process. If the two treatments are the same then they both have a probability of 0.5 of accounting for a death, however as they become different, the probabilities are divided differently. This probability is different from the *actual* probability of a death in the two treatments, for instance, both the treatments could have a death probability of 0.4. Assuming the actual probabilities of deaths in the two treatments, one could then compute the probability of an observed distribution of deaths in the two treatments. Since we assume this to be a Bernoulli process we can compute the probability of the two kinds of error (e.g. assuming the treatments to be the same and then computing the probability of the the observed distribution of deaths, and repeating it with the different value of the probabilities) thus one resolves the problems of

computing the variance, which is implicitly contained in the specification of the death probability and assuming the clinical trial process to be a Bernoulli process. process. This method has the advantage that the doctor just has to specify the expected number of deaths in both the treatments, in the case when they are the same and in the case when they are different. The disadvantages are that this method is too simple. Firstly there is no simple way to guess the number of deaths in a treatment (especially when they occur due to causes different from the disease as discussed later). Secondly this method doesn't take into account the fact that it is not always the death as an endpoint that we are interested in but *duration of survival*. There is no easy way such a simple method can deal with the hypotheses when they deal with duration of survival as a continuous parameter, rather than death as a point event. This is a very serious issue because survival as a parameter is quite crucial to studies. Thirdly this method also doesn't have a detailed model of the trial as a sequential process where patients are accruing continuously and usually have a post accrual period where they are just observed. Finally, it doesn't take into account the fact that in typical studies patients are stratified to ensure equal proportions of patients with different risks in different treatment groups. The usefulness of the model is restricted to small scale pilot studies or studies which are of very short duration, have a clear point event of interest and have a relatively homogeneous patient population. For most large trials which run for years one has to address more involved questions which are dealt with in the model discussed next.

Bernstein and Lagakos discuss a detailed statistical model of the trial in [Bernstein78]. They basically assume the process of disease to be an exponential process with the survival curve being an exponential one. Thus they deal with survival in terms of half lives. Their model also deals with the sequential process of patient accrual, post accrual observation, unequal allocation of patients to the two treatment groups and most importantly, with the stratification of patients.

Their model assumes a uniform distribution of patients who arrive randomly during the period of accrual and hence takes into account the fact that treatment is begun at different times for different patients. After the accrual period is over patients are observed for a certain length of time. While the patients are being treated, their survival probability is governed by an exponential curve which is derived from an informed guess of the estimated half life of patients in each of the treatment groups. Typically the half life is known for the control treatment which is a standard treatment and it has to be guessed for the new treatment, for which a small amount of data is

available form pilot studies. Before we relate this model to the hypothesis testing model we need to talk briefly about why stratification is essential and how this model deals with it.

Typically when patients are enrolled into a study they are randomized into different treatments. However it is possible (for example) that one of the treatment groups may have more patients from an older age bracket. This means that at least partly the number of deaths in that treatment group will be attributable to the greater risk faced by the patients due to their age difference. In general if the patient population is composed of different risk groups (e.g. older people, younger people, those with other disorders/diseases) they should be independently randomized into the two treatments to ensure that roughly equal number number of people from each risk group are assigned to each of the treatment groups so that any difference in the treatments is due to an actual difference in the treatment and not due to the bias introduced by a non-random distribution of patients with different risks. However a uniform distribution solves only part of the problem. What we are interested in are the deaths caused *only* by cancer and not by other causes. However in general it is hard to separate the risk faced by the patient due to the cancer, due to the toxicity of the treatment, due to the other causes (old age, disease etc.) Therefore the difference in the number of deaths which results from having faced a different risk corrupts the statistics, introducing *noise*, while we are interested in the signal, namely the difference in deaths caused by the cancer alone. One way to raise the *signal to noise ratio* is to compare each stratified group in each treatment to the corresponding group in the other treatment and then combine their results. This way one could give different weights to the results from different stratified groups depending on how accurately they represent the effect of the therapy as compared to other random causes. In general those strata which have a high risk to begin with will have less weight than those with lesser risk.

The Bernstein and Lagakos method (used in this project) stratifies patients according to different *hazard rates*, which are the parameters in the exponential curves (related inversely to the half life). Further instead of considering the ratio of the number of deaths in the two treatments it compares the ratio of the *hazard rates*, designated to be delta. Therefore in the light of the earlier model, delta = 1.0 corresponds to the hazard rates being the same for the null hypothesis (half-lives are the same in both the groups) and delta = 2.0 (say) for the alternate hypothesis (the half-life in the experimental group is twice that of the control group). While the *hazard rates* are different in the two treatment groups it is assumed that the ratio of the hazard rates in

corresponding stratified groups from the control and the experimental groups is the same (This ratio is delta as defined above, also sometimes called the *ratio of efficacy*). This is a strong assumption and is discussed later. The variance of this delta is calculated by combining the variance of each stratified group in a weighted manner. The details of the algorithm are beyond the scope of the discussion here but it is enough to say that since the combined variance is the sum of a number of random variables one can invoke the central limit theorem for computing the combined variance from the individual variances, which are computed from the size of the sample in each stratified group and weighted inversely according to their hazard rates. The algorithm mathematically adjusts the weights so as to minimize the variance and get maximum *information* from the data.

To use the above model to compute the power of a test the doctor specifies the hazard rates (or half lives) of the different strata, specifies the accrual period, the accrual rate, the post-accrual observation period, and the ratio of efficacy. The p-value is generally assumed to be 0.05 so the main variables are the power of the test (1- error of type 2), the period for which the trial has to continue, the ratio of efficacy, and the number of patients accrued into the trial. Given all but one of them the other can be computed. Therefore one can find out the power of a test, given the rate of patient accrual, the length of the test and the ratio of efficacy or conversely find the number of patients required or the length of the test, given the rate of patient accrual and the ratio of efficacy and the power of the test. Using the algorithm, graphs can be drawn between ranges of variables, to optimize (visually) the power of a test or some other parameter (for example the length of the study or the number of strata)

4.5 Assumptions, limitations and power

We saw that with a more detailed model of the process of the disease the Bernstein/Lagakos algorithm can answer much more detailed questions, and for the simplicity it has, it is quite robust. What are its limitations? Firstly it assumes an exponential model of the disease process which is true for most but not all diseases. One can make a favorable case for the fact that the degree of accuracy that the exponential model has for most cancers is justified in the light of the fact that some of the other parameters such as the difference between the treatments (the ratio of half-lives, deaths etc.) can have errors of the order of 100%, so any additional error introduced by an assumption of the exponential kind is not likely to radically change the final outcome. On the other one could argue that the only difference between the Schoenfeld model and the

Bernstein/Lagakos one is that the number of deaths are calculated in a different way and then the hypothesis testing theory is the same. This is largely true as far as the mathematical algorithm goes, but *calculating the number of deaths is precisely the crux of the problem and any assumption regarding whether the process is continuous and/or exponential will crucially affect the accuracy of the hypothesis testing algorithm.* If the the variance is calculated with error, then the outcome will also be influenced. Other assumptions made in the second algorithm are that the variance can be calculated by using a normal approximation which is true for a) large samples (> 50) and a reasonable number of strata (~ 10). Further the assumption is made that the ratio of efficacy is the same across all the strata. This raises a thorny issue about the nature of the *noise* or the random causes. It is not clear whether the risk faced by the patients due to the cancer, due to the toxicities and due to other diseases/causes are independent or synergistic. Statisticians can think of them as being additive or multiplicative depending on whether they are independent or not. *This* algorithm assumes them to be multiplicative, so that they cancel out when computing the ratio of efficacy. There is no particular evidence that this is strictly true. On the other hand stratifying the patients in the first place introduces more accuracy and it is hard to decide whether the advantage gained by stratifying the patients is lost by these additional assumptions of normalcy and the multiplicative nature of noise. Another objection has to do with *guessing* on the part of the doctor regarding the ratio of efficacy. What do the doctors guess best? (the number of deaths, the hazard rate, the half lives?) Most of their data is gleaned from scant pilot studies or inferred from a combination of other studies. No data exists to show the preference of one kind of parameter over the other and doctors frequently guess what they are most familiar with (e.g. half lives), which may not necessarily be the best parameter to guess (in terms of the sensitivity of the error in the final outcome to the error in guessing)

4.6 Underlying models

From the above discussion it is evident that although the Bernstein/Lagakos model is better equipped to deal with the questions regarding the statistical design, it raises issues of its own. However some of the issues it raises arise from the nature of the domain (cancer) itself while others are specific to its method. For example assuming an exponential model for the survival curve is a specific assumption while problems of stratification and parameter estimation occur in all methods which are sufficiently powerful to deal with the problem in any detail. As far as the exponential model is concerned its main advantage is simplicity. There is no reason why one

shouldn't use more detailed graphical models of the disease when such information is available. The methods of computation are only slightly more detailed (a graphical integration rather than a closed from computation) because many more parameters are needed to specify the (say a piecewise linear) model, but the essential computation remains the same i.e. how many events of interest have occurred by a certain elapsed time of the study. Therefore it is not hard to extend the model to include arbitrary disease processes and the exponential model is a good starting point, especially in the absence of more detailed information. Different models lead to different variances of various parameters and it is usually in retrospect that one knows the true process of the disease/treatment but one has to commit oneself to one before one can design the trial. In some fortunate cases one can retrospectively correct the model and use the same data for a different analysis.

One additional but very important basic issue we haven't discussed is the issue of *statistical independence*. We have conveniently assumed that the two treatments are unrelated to each other. What if the new treatment is a variation of the older one? if one treatment is A followed by B and the other is just A, then one cannot assume them to be independent. The responsibility is up to the doctor to ensure independence before he uses the simplified model discussed above. It is not that it is hard to deal with common factors, just that one has to recognize them and use an appropriate model. And as long as one uses a specific model, one should take care to see if its assumptions are satisfied. Along with the statistical independence it is assumed that the population is *random*, i.e. that the results of the sample are a for representative of the entire population. It is again up to the doctor to evaluate the sample to see if it is not biased in any particular way and if it is, then to qualify the results appropriately.

4.7 Importance in terms of the real validity of the trial

If a treatment is like insulin for diabetes then it proves its validity pretty much on its own, because its dramatic impact will be evident even in poorly designed studies. However as is more often the case the improvement is less dramatic and if the study is badly designed then a potentially good direction for investigation might be lost. As discussed earlier the way a trial is designed the error of a false negative is considered secondary to the error of a false positive, in the interest of patients. If there is going to be more than one trial, the damage is likely to be less severe if there is a false positive because then other studies will soon point out the difference. However most trials are carried out only once since they require a sizable investment of economic and human

resources, therefore if there is false negative no study may follow it up. In fact it may be hard to establish whether a negative is a false negative when there is only one trial since the whole science of statistics is based on sampling and if there is only one sample all that can one say is "If there are 100 similar such studies then the likelihood of .... is such and such" but there will not *be* any more studies. Therefore if one is going to design a trial of which there is going to be only one of a kind then one should take special care to avoid the false negatives. A trade off has to be made between the cost of the trial and the potential benefit/harm which might occur due to an insufficient trial. Another way of saying that is: Not only should the trial guarantee the power of the test in case the hypothesis is true but also have information about other viable hypotheses in case it is false, such a power being acquired at a cost of a larger trial, more money and time.

4.8 Implementation

The statistical algorithm (the Bernstein and Lagakos one) has been embedded in an interactive program. When given some specific parameter to be computed, it first determines the values of the parameters needed to compute this particular one. If the user doesn't know the value of the needed parameters directly, the program tries to compute it indirectly from its knowledge of the dependencies between the parameters and treating the problem of finding the needed parameter as a sub goal or a sub computation. As in the rest of the protocol writing program the data is checked to be within reasonable bounds and a facility exists to specify the data in any reasonable units.

Apart from being able to do point computations of the kind discussed above, the program can plot the effect of varying an independent parameter on a dependent one (such as seeing the effect of a varying number of patients on the chance of detection).

4.9 Summary

We have described in this chapter how the assumptions of different models of the disease and the trial process can potentially lead to different conclusions. The main point of the above discussion is to separate two kinds of issues : The theory of hypothesis testing and the parameters required for it, and the different models of the disease and the trial as an experiment. The theory of hypothesis testing is the most uncontroversial part of it, because it is simple and depends on relatively few parameters. It is in choosing the different models and seeing to it that the different

statistical assumptions are satisfied or not, that the major problems lie. Different models give different inputs to the same theory of hypothesis testing and can lead to different decisions. The two main guidelines for choosing a model are : To see that it describes the disease process and the trial process faithfully enough and in enough detail that it answers interesting and important questions, and that the different statistical assumptions are satisfied so that the answers are dependable. Most of the expertise and art of choosing a model goes into seeing how faithfully the model describes the disease and the trial process and what are the relative tradeoffs involved. A more intelligent system must have a number of models and should be able to choose among them, depending on the amount of information available. Also a more intelligent system should be able to have a more detailed knowledge of the treatment to be able to verify the assumptions of statistical independence. The program is interactive and includes the ability to plot graphs between parameters.

5. How it works : Nuts and Bolts

This chapter describes how the system is configured and illustrates a step by step instantiation of a small part of the protocol. Essentially the system is an interpreter operating on a data base. The following sections describe in more detail the control structure of the interpreter and the organization of the data base, followed by the example. The rest of this chapter describes the "utilities" that enable the user to have control and flexibility in using the program.

5.1 The Interpreter

The basic design of the system is shown in fig. 3. An instantiator program "reads" the generic structure of the protocol from the data base and instantiates a specific protocol based on the user responses to the queries posed by the program through a user interface. Corresponding to each node of the generic protocol the program makes an instance node for the specific protocol which inherits the properties of the generic node and may further contain individual properties to "specialize" it. In this way a number of protocols can exist which all share common knowledge and yet are unique because of their specific differences.

As the protocol is being incrementally instantiated the program checks each of the input data for consistency (whenever that is possible to do that at input time) and asserts the data into its representation of this individual protocol. This can include simple type/range checking to more complicated constraints (discussed later) which refer to other parts of the protocol. Whenever a particular section of the protocol is complete the program then applies the constraints that relate the particular part with other parts of the protocol which are complete.

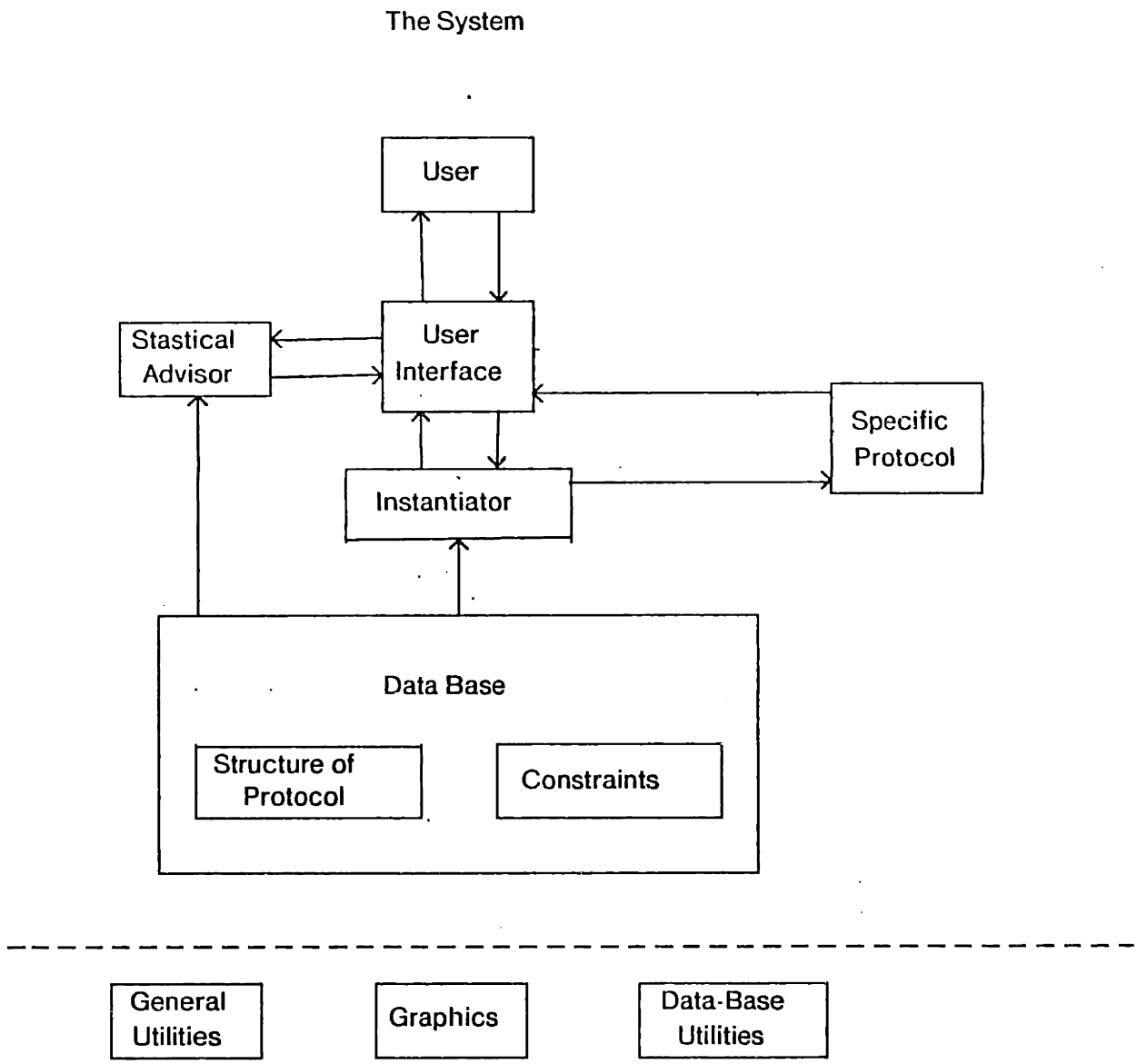
The simplified central instantiation algorithm is as follows

1. Read the node and determine if it is required to be instantiated.

   If it is to be instantiated then get the data for the node as specified by the node itself.

   Once the data is acquired apply the constraints on it to determine if it is valid. If certain constraints cannot be applied then defer them and note the fact. The constraints are supplied by the node itself.

Fig. 4. A Schematic Description of the System



Based on the data, propagate any constraints that might exist.

Instantiate any inferiors that the node may have, otherwise move on to the next node.

Nodes can be of two types *object* nodes or *property* nodes. When property nodes are instantiated they just attach a property to their superior node while object nodes create a new object. For example the node [name drug] is a property node and will attach a property by the name "name" to the superior node (in this case [drug treatment]) which must be an object node. The data required by the node may be of different types such as a number with units, a choice of one response out of many, or free text. Each of these data types is subject to checks in the form of constraints. The node itself supplies the data type and the constraints. The information supplied by the node is of three types

a) the control information for the instantiator (whether self instantiating or not, what type of data)

b) specific data to be used for instantiation (questions,possible answers, how many instances are possible)

c) Constraints and deamons.

For example consider the following node, in which the names beginning with & are property-names and the value(s) following them are the values assigned to the node under that property-name. For instance "input-line" is the value of the property "data?".

```
[name regimen
  &data? input-line
  &prop t
  &question '|Enter the name of the regimen to which the patients are
              assigned after randomization :|
   &check (check-if-not-repeated
            (context-name '[randomized-regimens])
                          answer
                          'name)]
```

In the above node it is indicated that the data is a text typed by the user and after the name is typed it is checked for the fact whether the name has already been used before by looking at all the regimens in the context of the current protocol. To look at another example:

```
[subtype lung cancer
       &data? choice
       &question '|What is the type of carcinoma?|
       &poss-ans '(epidermoid large-cell small-cell adeno)
       &tasks (if (same answer 'small-cell)
                  then  (make-req '([cytology lung]
                        [confirmation lung-cancer]))
                        else (not-req '([sputum-cytology])))))]
```

In the above example we see that the data is going to be one of the known categories of lung cancer and based on the response appropriate constraints are propagated. For instance once it is known that the type of cancer is small cell carcinoma then the cytology of the cells is required to confirm the cancer and the appropriate details of the test will then be asked of the doctor. Also if it is known that the carcinoma is *not* of the small cell type then a certain kind of cytology (sputum cytolgy) is not required so no questions need be asked regarding that.

The advantage of having a simple central instantiator is that since most of the information is supplied by the node itself, as in a typical data driven system, it is very easy to change the structure of the protocol, namely by changing its *description.* Although most of the protocol is instantiated by the simple instantiator there are parts of the protocol which are "self instantiating" or in other words are instantiated by special functions which are required because of the parsing required in certain parts of the protocol. While the basic description language is sufficient to describe the complete protocol, these special functions just help in acquiring strutured data (such as in a tabular form).

While the protocol is being instantiated the user has the control of changing the order of instantiation of various parts of the protocol, looking at the parts of the protocol already completed, and has sufficient control that he can abort, save and continue the protocol as explained in more detail later in the chapter. The statistical advisor is implemented as a separate utility although it makes use of the same data base.

5.2 Brand X

The program is written in Maclisp and the data base is represented in a representation language called "Brand X". It is a language to implement semantic networks. It is has features which make it easy to build one's own representation language. The details of the language can be found in

[Szolovits80]. In this section we will just briefly review the basic features of Brand X so that the data base can be described.

Brand X is implemented as an extension of LISP with the following new features

a) Unique and canonical structures In Maclisp only atoms are interned so the only way to refer repeatedly to the same structure is to bind it as the value of and atom. In Brand X the concept of uniqueness is extended to include lists. Along with lists there are *unique lists* in Brand X. If one constructs a unique list (ULIST) then one can refer to it again by reconstructing it from its parts. This ability to reference unique structures is implemented by interning unique lists. This ensures that if one uses the same parts to construct a list then one gets the same list back

```
(eq (list 'a 'b) (list 'a  'b )) = => nil
(eq (ulist 'a 'b ) (ulist 'a 'b) ) = => T
```

ULISTs in Brand X syntax are represented by square brackets

```
(ULIST 'a ' b) = => [a b]
```

so that the above lisp expression can be rewritten as

```
(eq '(a b) '(a b) ) = => nil
(eq '[a b] '[a b]) = => T
```

Having unique lists has a direct bearing on the naming problem in semantic networks (which we discuss in greater detail later) because we have the power to use expressions as names (since they are unique). This makes the names more meaningful since they can be composed of meaningful sub expressions which define the context, and at the same time makes it easier to reference them by just reconstructing them by using their sub expressions. In general unique lists can have non-unique subexpressions. Those unique lists which have only unique structures as their subexpressions are called canonical lists. For example [(a b) c] is unique but not canonical. Its canonical version would be [[a b] c]. It is clear that if we are to use the ability to reconstruct names from their subexpressions then we will be using canonical lists exclusively. This is because the only way to recreate a unique list with non-unique components is to keep track of the non-unique parts by attaching them to atoms, in which case one may as well canonicalize them.

b) Universal property lists
As compared to Maclisp where only atoms can have properties, in Brand X *any* object can have properties. This is especially useful because it would not be of much use to use expressions as names for

reference unless one could assign properties to them.

c) Labels
Providing a convenient abbreviation to refer to complex expressions

d) An extended lisp notation
Brandx provides a very convenient and systematic facility to read/print Brand X "assertions" in a compact syntax. This makes it possible not only to program more efficiently because it is syntactically more compact (and thus makes it easier to specify or input a semantic network easily) but also makes it possible to save new structures by just printing them. One should also not forget the invaluable ability it gives the programmer to "look" at the data base in a compact, coherent way rather than have to look at the properties of individual objects. The BrandX printer/reader is sufficiently robust to read/print circular structures.

e) triples
Brand X also supports triples, however we will not elaborate on them since they aren't used in this project.

5.3 The Data Base

The data base consists of the structure of the protocol, disease related knowledge, knowledge about drugs such as their spectrum of effectiveness, their typical doses, their typical toxicities. It also contains knowledge of possible toxicities such as their causative drugs, their grades of severity, palliative measures and the way they change in time. Also in the data base is the "general" knowledge such as conversion factors between various units etc.

Since it is a data driven system, most of the information about the protocol is contained in the data base hence the organization is detailed, complementing the corresponding simplicity in the central instantiator.

5.4 Organization

Most of the knowledge is organized by context using ideas of generic and specific concepts, as suggested by Martin [Martin78]. Concepts are represented as Brand X Ulists, with their links to related concepts being expressed through properties. When a particular part of the protocol is being instantiated then the local knowledge required for that part is derived from the properties of that particular node and the nodes related to it, as well as by the general knowledge

data-base which is common to all the nodes. The knowledge is organized using hierarchies wherever possible. Although it would be expected that such an organization would always significantly simplify representation greatly, the difference is dramatic only when the domain has a rich internal structure (such as the real world). The representation of knowledge in this project is governed strongly by the structure of the protocol and that is a natural abstraction around which to organize the facts rather than in a more general framework. Ideas of frames as discussed by [Minsky75] have been used. The basic strcuture of the protocol is described by a tree structure consisting of a relationship of *roles* for example

```
[therapy &roles [stratify] [randomized] [regimen]
                        [dose-modification-rules]]
[stratify &roles [criteria]]
[criteria &roles [category]]
[randomize &roles [randomized-regimens]]
[regimen &roles [treatment-plan]]
[treatment-plan  &roles [treatment] [end-test]]
[treatment &roles [name treatment] [treatment-type]
                              [detailed treatment]]
[end-test &roles [duration-before-test] [test-cond]]
[test-cond &roles [test]]
[test &roles [test-type] [poss-response]]
..... ..... .... ... ... ...
[dose-modification-rules &roles [toxicity-modification-rules]
                                [other rules]]]
```

Each of the above nodes in turn are described (recursively) in more detail, with the corresponding information required for instantiation. An example of the description is the node [name regimen] shown earlier in the chapter. Along with the structure of the protocol nodes and their descriptions the system has a lot of "general knowledge" and one example of that is:

```
[adriamycin = adriamycin
       &short-name ADM
       &common-mode IV
       &single-dose-limit (110.0 mg//sq-meter)
       &cumulative-dose-limit  (550 mg//sq-meter)
       &poss-toxicity &agent
            [nausea-and-vomiting 1
                    &instance-of [nausea-and-vomiting]
```

```
                &seriousness mild &frequency mild]
        [alopecia 1
                &instance-of [alopecia]
                &seriousness mild moderate severe]
        [cardiac toxicity 1
                &instance-of [cardiac toxicity]
                &seriousness  mild moderate severe]
..... ..... ..... ..... ..... ....]
```

The above description specifies the drug adriamycin by its drug dosages, its expected toxicities etc.

5.5 Constraints

Constraints are a natural way to represent dependencies between different parts of a system. These dependencies arise from requirements of consistency between related parts. It is these constraints that are responsible for the "intelligence" of the program and without which the program would be little more than a template for creating a protocol. The dependencies between different parts of the protocol (discussed in chapter 3) have been encoded in the form of these constraints. Along with the general structure of the protocol there are associated constraints which describe the relations between different parts of the protocol. Each node carries the constraints required for its instantiation. For instance in the treatment section along with the description of the drug dosage there is a constraint which specifies the bounds of the normal dosage (which can range from a single upper limit to a detailed time frame information). Whenever a certain node is instantiated its associated constraints are invoked. Local constraints are checked as the data comes in while the constraints which are to be checked against other sections may have to wait till the relevant data becomes available. Also as the nodes are instantiated the constraints are propagated to related nodes to reduce future decision making. For instance if it's known that Adriamycin is one of the drugs being used then there must be a rule for eliminating patients having cardiac problems in the eligibility section so that the doctor doesn't need to be asked whether such a rule is necessary or not.

5.6 Local and non-local constraints

Constraints which depend on just the node or nodes previous to it in the same part of the protocol are easy to apply because all the information required for their application is available, and in case the data is inconsistent, the user can immediately correct it. On the other hand constraints which relate information from different parts of the protocol can lead to problems because the different parts of the protocol may not be complete and the application of the constraints may have to be deferred. For example some of the constraints are defined in terms of the cycle length of the treatment and the cycle length itself maynot be defined at some point. This could occur for instance while defining the study parameters, in particular if we are defining the schedule of monitoring the wbc count, then we need to check whether it is consistent with the drug dose schedule so that the appropriate counts are available just before drug administration. However if the drug schedule is defined in terms of the cycle length which maynot be defined, the constraint has to be saved so that it can be applied later. This obviously necessitates having an ability to change things in case they are found to be inconsistent later.

5.7 Representation of constraints

The constraints are represented both declaratively and procedurally. As far as possible the constraints have been represented explicitly rather than procedurally. Constraints maybe as simple as numeric bounds on a parameter or they may be very complex requiring special functions to relate different parameters and do the checking. For instance checking if the toxicities due to different drugs overlap requires an explicit simulation of the toxicity characteric and has to be done by a special function. On the hand constraints which just define inequalities in terms of other parameters can be expressed in terms of simple language which can be interpreted by a corresponding evaluator. In general one needs at least a vocabulary of higher level primitives than bare lisp in order to represent constraints uniformly. An example of the use of simple constraint vocabulary used in this program are:

(if <predicate> then <action>)

the predicate can be logical/arithmetic, or may check for existense, or relationship of one node to another. An action can be an error situation, make other nodes required or not required, instantiate other nodes, defer constraints.

(operator [node1] attribute [node2] attribute)

for instance

```
(if (abs> [drug dosage] value  [drug] single-dose-limit)
  then (return error))

(if (part-of [drug 1] [treatment 2])
   then (instantiate [toxicity-mod-rule [drug 1]]))

(if  (same answer 'small-cell)
   then (make-req  '[cytology lung]))
```

where abs> is a predicate which compares values in different units. Other preciates can be abs>, abs = , check-range, contained, not-repeated, required.

While in principle it is desirable to represent most facts and constraints in a declarative form, in practice this is not always possible or feasible. This is because a declarative representation requires a declarative language and its interpreter, in terms of which the particular knowledge can be described. This may be difficult to begin with and unnecessary at times when it is just sufficient to express a certain constraint procedurally. For instance while inputting numbers for dose modification rules it is important that the categories of toxicities not overlap and that they be contiguous and should cover all (reasonably) possible ranges of the toxicity. It is easier (and sufficient) to have a function which can perform arithmetic tests which implicitly contain the above requirements rather than describe these constraints (in a logical and topological language) and then "apply" them with an evaluator. Also certain kinds of constraints may require specific computation (such as simulation to detect the overlap of toxicities) which is hard to capture in a structured manner without a comprehensive effort to implement the required set of primitives (such as those necessary to describe process theory). The situation is not always that clear however. In case of very specific bits of knowledge and variations, it is tempting to provide just a special purpose solution for that particular case. This results in a system which is not flexible because it doesn't have a general framework and yet such a trade off must be made if a real project is ever to be completed.

5.8 User Engineering

It is not an accident that in the human brain one third of the cortex is devoted to processing the vision, auditory and other sensory inputs. The I/O and other parts of a system that interface with the environment are at least as complex as the central units if not more. More often than not while the central processing is done by elegant structured algorithms, the I/O interface doesn't lend itself as easily to generality because the programmer has more control over the internal representation of the the problem in terms of which the absolute "information content" is less complex, than the enormous amount of information required to do displays, formatting and specialized parsed inputs. It is not surprising therefore that almost half of the programming effort in this project has been devoted to implementing the user engineering utilities and special I/O frills in order to make it more useful and friendly. They include parsing the input of the protocol in order to reduce excessive questioning, especially when the data has a known structure, displays of the parts protocol already instantiated, ability to read, examine and continue earlier protocols, rudimentary change and explanation facilities. The next sections describe each of the above features in more detail.

5.8.1 Parsing input

Most of the protocol is instantiated by a simple control structure which looking at the generic protocol, and instantiates each node recursively, where each node brings its own information necessary for instantiation as in a typically data driven system. This approach although sufficient is quite verbose and lengthy when a number of successive inputs are in a structured form and the number questions need be asked can be reduced, taking advantage of the structure. Therefore to facilitate the input of these parts control is handed over to special functions which instantiate that part with a knowledge of what is going to come so that most of the data can be parsed as it is input in a structured manner (such as tables of schedules and dose modification specifications).

Chapter 2 illustrates the structured input for many parts of the protocol design such as the design of the detailed treatment part (the specification of the drugs and their scheduled, the rules for modifying drug dosages and the specifications of the study parameters as also the specifications of the categories of stratification. In general there is no way to do the input parsing except set up separate finite state algorithms to deal with each of them. At the same time it has to simulate the instantiator, whose function it takes over. Therefore care has to be taken to have it

instantiate nodes with the same conventions and checks etc. as the central instantiator. This part also has to keep a buffer of the input so that if an error is made then the input can be displayed in the same form up to that point.

5.8.2 Printing

Once the protocol has been instantiated, it has to be displayed and printed in a format most familiar to the doctor, i.e., as printed in a typical protocol. A special set of display functions, one for each section, print the information from the internal representation in a format almost exactly as if it were part of an actual written protocol. Parts of the current protocol can be examined separately. Earlier protocols can also be read in and examined in a similar manner. The conversion of the display to an actual hardcopy is simple enough in most cases where a left to right line by line printing is involved. However when the graphical output uses the two-dimensional instruction set of typical display terminals, the whole display must be buffered for printing when it is complete.

5.8.3 Displaying intermediate forms during aborted answers

The functions mentioned above are able to display only completed parts because in order to be efficiently programmed they assume the existence of complete substructure. It would too much decision making to check for the existence of each subpart and then display it. For this reason displays of forms which are being input in a structured manner and have to be aborted or continued at a point are managed by keeping a record of the whole input for that structured part of the protocol (such as tables etc.) and redisplaying the buffer up to that point.

5.8.4 Explanation

The level of explanation is rudimentary in most places where a "?" will give a canned explanation of what is expected at that point. Somewhat more is explained in the statistical section where definitions and interpretations of the the various statistical parameters can be found. Similar definitions and explanations can be added for any part of the protocol because the control structure already exists and only information need be added to the data base. However there is no facility at present to incorporate explanations of the "Why" or "how" kind which involve a knowledge of the goal structure as discussed elsewhere in this thesis.

5.8.5 Change

The general problem of change involves being able to change any part of the protocol while one is instantiating it. This turns out to be hard because it involves the updating of a constrained data base, as discussed in the chapter on "Computational considerations". Therefore only a limited change facility is provided where at the least while in the process of instantiating the protocol one can abort something present and go back to the previous step. Rather than have the user change anything in general, a set of elementary "change operations" are available such as adding removing a drug, changing eligibility conditions, changing the specifications of the study parameters, modifying the treatment, changing the dose modification rules, changing the definitions of the ranges of the toxicity etc., in short almost everything important that the doctor might want to modify but not everything and not in arbitrary order.

5.8.6 Storing and reading

A complete facility is provided for storing current protocols and reading in earlier ones. A directory of protocols is maintained which lists protocols by name. Any number of protocols can simultaneously exist in the system and can be independently examined. They are displayed in the format of an actual protocol by the display facility mentioned above. The implementation of the storing and reading are facilitated greatly by the Brand X reader and printer which are the backbone of the the storage and retrieval system. The special problems encountered are described in the chapter on "Computational considerations".

5.8.7 Continuing

It is possible to read in an earlier protocol and continue from where one left off provided one interrupted the protocol in an appropriate way as defined in the section on "Aborting and interrupting" in chap 6. Essentially it requires that one exit at a stage when the instantiation of particluar node has been completed. With this restriction one need not store an extensive amount of information in order to start where one left off, which would have been the case if one allowed interruption (for example) in the middle of a a constraint check or a query.

5.8.8 Adding to the data base

Most important additions of "knowledge" in the form of constraints are already programmed into the system and require the intervention of the programmer to add to it. However many mundane changes which require the addition of information such as adding a new drug, defining a new toxicity, adding an explanation, can be handled by system defined functions thereby enabling the doctor to modify parts of the data base in a consistent manner. In a similar manner some changes to existing information can be made such as redefining the definition of toxicity, changing the value of some limits of certain variables such as the maximum dose of a drug, or the normal range of a parameter etc. The changes are stored in an incremental data base which is always loaded to update the information. From time to time information in the main data base is updated directly from the incremental data base by manually replacing the old information by the new.

6. Many Problems, Some Solutions

We have now seen enough of the structure of the protocol, the interpreter and the data base to be able to understand some of the more involved and fundamental problems. Some of these originate from the state of the knowledge of cancer and the difficulty in formalizing and representing it. Others derive from a computational perspective and are independent of the domain.

6.1 Problems relating to knowledge and its representation

In order to check the protocol for different kinds of mistakes and omissions the program must have a representation of the structure of the protocol and the disease related knowledge in terms of which a new, hypothetical protocol can be designed. This knowledge consists of information regarding drugs (e.g. their typical dosages, modes of administration, common toxicities caused), disease related knowledge relevant to formulating eligibility rules, and knowledge of the treatment plan itself and what constraints it must satisfy to be a protocol free of incomplete or inconsistent specifications. It is clear that unless a certain part of the protocol is "understood" (i.e represented along with its associated constraints) the program cannot detect any errors in it because it cannot reason about it. The problems regarding formalization and representation of the required structure and knowledge arise from a number of sources such as inadequate knowledge of cancer itself, subjectivity in medicine and variations within protocols. These issues are discussed next.

6.1.1 Lack of knowledge of Cancer

The state of the art of medicine in the field of cancer lacks fundamental knowledge of disease processes and hence knowledge of how various therapeutic agents work. This results in design of therapies at an "informed guessing" level and it is an exception rather than the rule when one finds a formal causal justification for the choice of a particular treatment. The few mechanisms of drug action which are understood at a biochemical level are so complex that nothing short of a large complex simulation can make such understanding useful. Even principles of pharmaco-kinetics -- how the drug is metabolized by the body -- and concepts of "cell kill strategy" are not understood at a level of certainty and simplicity that they can be formalized and used by a computer program to test the "optimality" of a particular therapy. Such an undertaking

is enough to warrant a full scale research project and is certainly out of the scope of this project. This limitation forces us to deal with the therapeutic and toxic effects of drugs in a much simpler way and we are not able to deal with the "Intent and Objective" section of the Protocol at all since they attempt to describe the philosophy behind the therapy in a non-formal way. This means that the program does not "understand" the higher level rationale behind the treatment and hence cannot check for the consistency of the treatment design with the intent of the study and related questions. What is needed is an over-all model of the treatment which will take into account the *intent* of the protocol which influences specific decisions about the the treatment. For instance, a much higher level of toxicity may be acceptable in a curative protocol where the risk is worthwhile to achieve a possible cure than in a palliative protocol. Without a comprehensive model of the intent of therapy it is hard to detect undesirable levels of toxicity since they may be allowable depending on the context. What data one needs to observe may also depend on the intent of the study. For example, study parameters to detect response, to control toxicity, and to study the progression of disease derive from an over all model of the disease and also determine the frequency and the timing of the observations. By an overall model here we mean a model which includes the model of the clinical trial and a model of the disease process itself.

6.1.2 Specification of rules in a descriptive language

Although many measurements can be made objectively, still many others require a subjective evaluation. While an effort can be made to make the descriptions of such measurements as specific as possible, subjectivity is still unavoidable. Such situations arise while describing the extent of disease (for eligibility), while describing criteria for grading symptomatic toxicities (e.g., vomiting) and while describing rules for judging to what extent a patient has responded (what does a "partial response" mean?). Such rules by necessity are descriptive and this makes it very hard to formalize such descriptions as is evident from the following examples:

> The tumor must be clinically and pathologically stage I or II bronchogenic carcinoma. Appropriately identified hilar and mediastinal lymph nodes must be biopsied and submitted for histopathologic review even if they are negative at the time of thoracotomy.
>
> Uncontrolled documented severe bacterial infection will be cause to withhold chemotherapy until it is controlled.

> Vincristine will be omitted from the any cycle if muscular weakness becomes disabling. Central nervous system toxicities will be evaluated as to cause and if related to drugs, they will be stopped.
>
> Methotrexate should be stopped if the patient has more than 3 oral ulcers or is unable to eat.
>
> Definition of Partial remission for evaluable non measurable lesions: A definite decrease in the size of diseased areas amounting to an estimated 80% regression ("close to complete regression") or better. This should be confirmed by at least two investigators evaluating independently, or photographs of x-rays should be submitted to the study chairman for confirmation.

There is no recourse but to provide a template kind of structure for the more manageable kind of rules, but that only solves the "text input" problem. The rules are not "understood" by the program and it cannot reason about them. In this way the knowledge and the semantics is abstracted out of the system and one is left with a part of the system where the programmer has the responsibility of ensuring it's correctness.

6.1.3 Need to describe spatial knowledge

We earlier referred to the difficulty in describing the location and the extent of disease, based on the subjective nature of such descriptions. Another related problem about such descriptions is that they require descriptions in terms of the anatomy of the body and assume a common sense knowledge of space. For instance if the program is to be able to check a description which says that the tumor should be "above the diaphram, attached to the chest wall, on the right side" etc. then it has to understand and reason with spatial knowledge to decide if the description is possible/ambiguous/incomplete. While such spatial knowledge and reasoning associated with it are interesting problems for AI, the effort required to incorporate them is beyond the scope of this project.

6.1.4 Multiple models of drug administration

It is possible to check a certain drug schedule for errors if it is known what is the "correct" dosage/way to administer the drugs. This depends both on the particular nature of the drug and the rationale of the treatment. For instance a drug may be administered in a single dose or over a period of time from a few hours to a few days, with different therapeutic/toxic effects, not all of which are predictable, depending on whether the drug is cell cycle specific, what are it's

pharmaco-kinetics and half life in the body and whether the treatment is curative or palliative. The same drug may be used in different treatments, in combination with other drugs, without a detailed justification of why it is given in a certain way, the most common justification being that it has worked in the past. If the study is to be comparative, then some factors have to be kept constant and then one cannot alter the drug administration schedule. Because there is no universal definition of correctness, it is difficult to determine if a suggested dosage schedule is acceptable. Instead, the program can perform only simple checks based on the expected concentration of each drug over time as predicted by the dosage schedule and the drug's pharmacokinetics. This prediction can shed some light on expectable toxic responses, but has little predictive value for expected therapeutic effect.

There not being a particular "correct" dosage or way to see if a new suggested schedule is allowable, one can just perform simple time frame checks based on information regarding the concentration and frquency of administration of the drug and it's half life in the body, thus shedding some light on the toxicity, but not on the therapeutic effect.

6.1.5 Variations Within Protocols

Although one can formalize the structure of a protocol in a general way, individual protocols still have sharp departures from the usual procedures. For instance although most studies would require dose modifications in case of severe toxicity a new study involving the effects of an "intensive high dose regimen" may have the rule "No dose modification rules will be in effect for the first 2 weeks of the treatment". Such a rule derives from the high level rationale behind the treatment and since that is outside the scope of this project it is difficult to deal with such rules except to have a template structure for such unusual categories of rules. In the absence of a fundamental justification each user has to specify explicitly whether he wants such a rule or not. The more the program "understands" the fewer unnecessary questions need to be asked of the user. For instance while specifying the drug schedule if the doctor doesn't specify the cycle length then the program automatically asks him to specify the repetition time (e.g q 3 weeks) independently. From the structure of the protocol that is already complete the program should be able to deduce relevant facts and automatically "fill in" related information (by propagating constraints). This depends on (1) How much information is available (state of the art) and (2) How much of that information is formalizable for the purpose of representation.

6.2 Computational considerations

6.2.1 Storing

Once the protocol generation process has proceeded for a while, there are changes in the Brand X environment which represent the newly instantiated protocol. It is useful to be able to save these changes so that at a later time the protocol can be continued or just displayed again. In order that the newly instantiated protocol be stored it has to be separated from the generic protocol knowledge that it is attached to.

In order to save the protocol we need to write the Brand X structure of the protocol onto a file. In principle that is easy because the Brand X printer recursively prints out all the linked substructures and hence just printing the top node should save all the information in the protocol. The problem is that the data base is very strongly connected and since the instantiated protocol is linked to the generic one through an instance-of link, the printer recursively prints out the generic protocol and all its properties and (in short) the whole universe. There is nothing necessarily wrong with that except that the generic knowledge is redundantly stored with each protocol. One obvious and general solution is to write another printer which "knows" which nodes are generic and prints those nodes just at top level. That involves writing a special Brand X printer which does the type checking for each node. The solution implemented in this system is not general but suffices for present needs. It makes use of the fact that the only links which exist between the generic objects and the instantiated objects are the instance-of links. Since there is a uniform naming convention for naming instances which specialize everything, it is easy to construct the generic nodes from the instance ones. Therefore, since the instance-of link is encoded syntactically there is no need to print that link and hence when storing the instantiated protocol these links are deleted and reconstructed when it is read back again.

A directory of protocols is maintained which lists them by name and root node. To read a protocol back in one just needs to specify the name of the protocol.

6.2.2 Continuing

The process of the instantiation of the complete protocol can be long and tedious, involving a number of details which the doctor may not have available at the moment. It is convenient to be able to interrupt and continue at some later point.

If the state of the system was to be preserved completely there would be no problem, one would just pick up from where one left off. However the transition from one session to another session is accomplished by storing the partially completed protocol when the first session is over, and reading the stored protocol back into the system in the next session. Some information is therefore lost because the stored protocol is a representation of only the important completed parts of the protocol and their values. It does not, for instance, have a description of the many "system variables" which keep track of different processes. The "context" is lost and must be recreated to the extent that is required by the generality of the facility to continue.

Among the variables which keep track of the processes of the system are the ones which keep track of global information such as how many instances of a name have been created and thus have to be remembered for the course of the whole process of instantiation. There are other variables however which are temporary in the sense that they hold a temporary context which is important only for the instantiation of a particular node and once that node is instantiated, they are thrown away. If care is taken to allow the interruption only when nodes have been completely instantiated, then one may only need to store the global variables and not the many which hold temporary dynamic information which will soon become useless. This is the approach taken for this implementation so that one can terminate a session only if one has finished dealing with a particular node completely. This means that one answers the question posed by that node and also allow the completion of propagation of constraints and the "deamons" which side-effect the other parts of the protocol. Whenever the central instantiator is doing the instantiating, this is achieved automatically because it "knows" when it is not done with a node and wouldn't process abort or other commands for terminating the session. When control is handed over to node specific functions which instantiate the node then the function has to send a message to the instantiator which tells it whether the instantiation was completed or not, in other word whether the exit from a lower level to the top level was "proper" or not, in terms of not leaving partially instantiated structures at levels lower than that of a node.

Legal instances of exiting would be for example in the middle of the eligibility section when one has answered the question regarding the age of the patients and is about to move on to a question regarding the extent of disease. At this stage the previous node has been completely instantiated but the new one hasn't begun so it can be "wrapped up" without loss of "what to do next". However if one is in the middle of specifying a table of dose reductions and one exits out of it (for example in the middle of a question), there is no easy way to store the intermediate state of what was the partially specified dose reduction and carry on from there. Therefore one will have to wait till some "unit" of the table or a node is complete and control is handed back to the instantiator which is then ready to "look" at the next node, before exiting. The point is to exit in such a way so that an excessive amount of information about the state of the control structure need not be stored.

With the restriction that one may exit only when a node is completely instantiated (implemented through scoping of the exit privilege) it is enough to store the superior node, the node to be instantiated next along with the global status information and the instantiated protocol. This information is sufficient to be able to continue in the next session, in a consistent manner.

One needs to add that along with the global variables which need to be stored there are also the deferred constraints for which the complete information is not available yet and the nodes which are to be optionally completed later (which have been saved from questions like "Do you want to specify this detail now?"), which need to be stored. For example if the cycle length is not specified at the begining of the treatment specification, certain consraints requiring the cycle length as information will have to wait till it is specified. For example it is necessary to check the drug schedule too make sure that toxicities do not overlap in a dangerous manner. However if the treatment schedule is specified in terms of the cycle length, then the value of the cycle length has to be known before the constraint can be applied.

6.2.3 Change

To make a change consistently means to be able to undo or retract the effects caused by an earlier specification of a value and create new effects required by the new value. This is simple only in the case where the specification doesn't influence (in terms of propagation of constraints) any other nodes. For instance it is no trouble to change the categories of stratifications or their ranges, since they have almost no effect on any other part of the protocol. On the other hand

withdrawing a drug and adding another is almost equivalent to doing a major part of the protocol all over again, because there is almost nothing that is not affected. Adding a drug means that the specifications of toxicities may change, the dose modifications and even existing toxicities will change because of the additional drug, the study parameters may change because one may want to monitor more parameters in a different schedule because of this new drug, and finally the eligibility part may change because the toxicity part may add some more restrictions. The obvious strcuture which comes to mind in ealing with such retractions is the dependency directed backtracking described by Jon Doyle in his "Truth Maintainace System" [Doyle79]. The essential idea is to store *justifications* of any facts which exist in the data base and use the justifications to express the dependency relationships of different assertions to each other. When certain objects or assertions are no longer valid, the dependency information is used to recursively retract all the assertions which are influenced by the facts which are no longer valid, and if there is no other way to justify the same assertions. While it may seem that a dependency structure in the style of the TMS may help retract and replace appropriate structures, the actual problem is more complicated. Consider the case where the dose modification rules are a result of a composite toxicity caused by two different drugs (their seriousness levels are merged). In this case withdrawing a certain drug means redoing the dose modifications. Similar "merging" or loss of specific information takes place when one specifies the study parameters and their timing. Therefore recomputation is essential when changes in the treatment are made and justifications have to be stored in enough detail to be able to pin point exactly what needs to be recomputed. Such a detailed justification doesn't exist in this program for every inference, especially when, as illustrated above, a number of steps of inference are combined, without keeping a step by step of evry inference. Currently we have an ability to:

Abort the present question, or change the response to it.

Provide a set of useful "change-actions" such as withdrawing a drug, adding another, changing values of variables (e.g. doses, schedules), changing eligibility conditions, among others.

6.2.4 Updating a constrained data base

When the protocol is instantiated, at each stage of instantiation constraints are used to check if the response to a query is "reasonable" or violates some rules or restrictions. When a change is made all the constraints relevant to the node being changed need to be applied again to check the validity of the new datum. However this leads to the problem of applying consistency constraints to a data base which is in a transient state and will "become" consistent after a few steps. This happens when one is trying to change a value which affects other values, which one also intends to change but will not get to before the constraints are applied; thus the system "thinks" that the present change is illegal because it is not consistent with the old values of the other variables, to which the constraint relates it. Therefore one needs to know when the data base is ready to be checked for consistency and delay the application of constraints till then.

6.2.5 Aggregation of changes

The addition/withdrawal of a drug can influence many parts of the protocol, for instance the part of therapy which handles the dose modification part. If a number of drugs are to be changed then it is better to change all the drugs and aggregate all the new toxicities (some of which may be common to the new drugs) before doing the dose modification part. One needs a concept of "boundaries" between which all changes are made and then aggregated before being "propagated" to the other parts. The boundaries need not be the same as the parts of the protocol, they come more naturally from the constraint structure of the protocol. This idea was expressed in Mike Bosyj's thesis [Bosyj76].

6.3 Summary

The difficult issues stem from two broad sources: (a) the lack of knowledge about cancer and difficulty in formalizing it, and (b) computational problems. While the latter are amenable to solution, though not necessarily easy ones, the former may require advancement in the state of the art of cancer therapy before comprehensive attempts at resolving those issues can be made.

Not enough is known about specific processes of the therapeutic and toxic actions of drugs and the mechanisms involved in the development and spread of the disease itself. This limits the extent to which one can have an expert system which understands the underlying philosophy of the treatment, which in most cases depends on heuristics derived from previous

clinical experiments and not from a detailed knowledge of the various mechanisms involved. The failure to incorporate the underlying model of the treatment makes it difficult to take into account those variations in protocols which derive from a difference in the *intent* of the treatment (e.g. is it palliative or curative?). Difficulty in formalizing the knowledge of cancer results from subjective definitions of terms, especially those involved in the descriptions of the extent of disease and the measurement of effect. The need to reason about naive concepts of space (as in giving spatial descriptions of the location and the extent of tumors) makes the problem even more difficult.

The issues arising from a computational perspective have to do with the user engineering part of the project. In order to be able to store, retrieve, interrupt, continue, examine, display and print protocols consistently, specific conventions have to be set up, which limits the extent to which the user has freedom in demanding arbitrary actions. The most difficult problem is that of being able to change the protocol in a consistent manner. A complete solution this problem has not been implemented in this project, so only certain basic change-actions are permitted.

7. Discussion

There are many things learnt from this project, some the hard way. they include (h)in(d)sights about a)selecting domains b) selecting the scope of the project c) selecting appropriate AI techniques to solve the problem.

7.1 Logic of approach

We will evaluate our approach in this project from two points of view:

The role of the protocol in making the clinical trial effective and

The specific approach taken in this project to deal with the limited problem that we have attempted.

7.1.1 What makes a clinical trial successful?

Consider the the different factors which are responsible for the eventual success of a clinical trial:

Rationale of therapy

Nothing affects the clinical trial more than the proper design of the therapy in terms of an intelligent design of the different drugs chosen for the therapy etc. If the therapy itself is non optimal then even if the rest of the "information gathering" parts of the trial work well the information is gained about something which will be soon replaced by a better therapy.

Statistical design

Next to the design of the therapy perhaps the statistical design is the most important factor in the design of a trial. If the number of patients for the trial is underestimated then one may not be sure of the result, if it is overestimated then the results maynot be available in time, and may waste resources. Since such design depends on many subjective factors, pilot study results etc. it is hard to be objective in as much as one is doing informed guessing (e.g. how different are the treatments?). To some extent the therapy design may influence the statistical design because the assumptions made by the statistician about the independence of many different agents have to be assured by the nature of the therapy, (for example how the agents are grouped) so the logical design of the therapy crucially influences the statistics.

The protocol

After the therapy is designed and the statistical design is complete it is the protocol, the written document, which communicates the design to the many physicians who might use it. The extent of its clarity,

completeness and consistency determines the compliance with the protocol in the therapy. Violations of the protocol due to any reasons corrupt the data for statistical analysis. It is clear that a good protocol can be instrumental in assuring the consistency of the actual therapy with the design. A uniform and unambiguos interpretation is necessary both for administering the therapy properly and generating reliable statistical data.

Compliance

Assuming the design is good and that the protocol is written in a clear and consistent way the final success of the trial depends on the integrity of the physicians in recording the data and following the protocol. Chance events such as patients dropping out of the study also influence the data though in a few cases corrections can be made for them. It is clear that one has little direct control of these factors.

Finally the data has to be analyzed correctly. If the protocol has been designed properly then one would assume that the analysis should not be a problem. However chance events such as patients dropping out and retrospective knowledge about the nature of the disease may lead to analysis of the results in a different way (sometimes with even different hypotheses for which the same data might be applicable). Also early data regarding the difference in the treatments may force the study to be stopped or modified due to ethical reasons and that may significantly compound the analysis.

With the above perspective about the relative importance of different factors we see that if an expert program has to be of major help to the physician in designing a protocol then it has to know enough to help suggest new therapies or to even choose optimal therapies from a number of them. As we have discussed before, the available knowledge of cancer makes that goal infeasible for the present and we have to be content with a much more modest scope which deals with the statistical design of the protocol and limits itself to detecting inconsistencies in the treatment design. Apart from the statistical expert which has significant knowledge and power to aid statistical design the rest of the disease-related section of the program reduces to contingency verification while planning a treatment -- which by itself is not an easy problem. In that role the importance of the program is that of an intelligent assistant and its significance in the over all clinical trial must be seen in that light. While it is easy to see a more sophisticated statistical expert system for protocol design, it is more difficult to visualize an expert program which is knowldegeable enough to significantly help a physician in optimal therapy design.

7.2 The domain of cancer

Research in the area of cancer has traditionally depended on "informed guessing" of the efficacy of different agents, therapies and it is a field where "experts" operate more with intuition than with exact knowledge. Enter the AI researcher. What does he find that is good about this domain as an AI research area?

7.3 Why is AI appropriate for cancer research

In part the answer must come from why AI is good for medicine in general: Because the rate of growth of information and its amount is too large for rapid and efficient assimilation and use of it and the computer can be a symbolic assistant. In particular the state of knowledge in cancer is such that there are few clear insightful and illuminating principles. Most of the knowledge of cancer is empirical and detailed theoretical mechanisms are conspicuous by their absence in the majority of neoplasia. The result is that there is a plethora of good heuristics which are combined with whatever little formal knowledge is available to produce a better therapy. In the absence of well defined knowledge in the field it is tempting to build "expert systems" to help organize and exploit what little knowledg there is. Besides as in other fields of medicine expert systems can be of important use in managing cancer treatment. AI can aid in decision making to choose the best treatment from among many for a particular patient, or to choose a different course of action after the treatment has progressed for a while. Hopefully with increasing knowledge of drug mechanisms and disease processes and pharmacokinetics, AI will be able to offer help in designing *new* treatments also.

7.4 Why is cancer a difficult domain for AI?

> AI can organize knowledge when there *exists* some. By knowledge in this context we specifically mean explicit models of different mechanisms which are responsible for the disease and drug processes at work. For the majority of the drugs in use and known diseases no detailed mechanisms are known so the question of representing them and using them doesn't exist.
>
> Again AI can organize knowledge when it can be formalized. While there *are* hueristics which experts use to design better therapies such as the knowledge of cell cycle specific actions of drugs, in the introduction of any protocol it can be seen that such "higher level rationale" of the therapy is mostly presented in an informal way. Detailed knowledge of

how the disease develops and progresses is poor to nil in most cases and there is little that is available as clear facts to be used in a deductive framework. While there is an explicit taxonomy of descriptions which classifies the observed classes of disease, there is little beyond that description which is of illuminating and predictive use so that one can plan ahead. Researchers combine mostly empirical knowledge from earlier studies in logical combinations and use them to design new studies. A typical justfication might say "Study A indicated that giving drug X in 200 % of its previous dose incresed the response rate by 20 %, this study examines the effect of giving a still higher dose spread out over a longer period."

Cannot AI help discover new therapies in the sense of planning experiments in the MOLGEN style? Yes and no. In principle yes and in practice no, at least at this stage. For a similar feat "operators" (in the form of drugs) for cell kill strategy will have to be understood at a much more specific level. For a few drugs some kinds of model exists at the bio-chemical level describing through equations how they operate on the DNA and other cellular structures. These equations sometime number as many as 20 simultaneous equations, many with unknown constants. If enough is known, then in principle one can simulate the effects of many drugs and see (for example) if they are synergistic or not [Jackson80]. In practice this is not feasible both because of the lack of information and because of the scale of the computation -- it would require a full scale project of its own.

Apart from the medical part of the design there is still the planning part of the therapy or the contingency verification part which has to ensure that there is a complete, consistent description of the procedures etc. -- the topic and scope of the thesis. While it is easier to use AI techniques for this part, even here the lack of detailed knowledge of toxicities and their relation to drugs limits the efficacy because the planning crucially depends on the nature of the disease and the toxic and therapeutic effects of different drugs.

While the above discussion sounds overly pessimistic it is a general perspective and is biased towards evaluating possible AI approaches to the *design* of cancer therapy. There are certainly some special areas in which a significant amount is known about particular disease/drug mechanism/metabolism and there is no reason why AI techniques cannot be of help in those specific areas. They can also significantly help in bringing non-medical expertise (such as statistical expertise) to the physicians.

7.5 choice of representation

From the discussion of the domain we move on to issues of representation. A few general comments are in order. While we have encountered typical choices in designing a system of representation, it should be kept in mind that our domain and the depth of representation being limited (in a sense to be discussed more precisely later) we do not have as "rich" a universe of concepts, for which we might have needed a more elaborate representation.

7.5.1 A few links can do a lot

One finds a number of representation languages which have different features which make them appropriate, flexible and sufficient, according to different criteria. How then does one choose a representation language?

My experience is: As long as one is working with a fairly abstract description of the domain and one is dealing with a specific domain it is to better to build one's own representation language. This is not to say that there is not a trade off between using an existing language and bearing with its "features", and spending effort to build one's own. If one has a particular domain then there are peculiarities which will force one to make specific decisions anyway (especially when one is using a relatively abstract model of the domain). Using any of the higher level languages will probably bind one to more than what one wants while giving less than what one wishes.

Of course there are guidelines regarding how to go about it so that one doesn't have to rediscover and reinvent all the known basic issues in representation language design such as inheritance, instantiation, naming, etc. It would be ideal if representation languages came implemented in layers so that one could start at the required level of "nuts and bolts" and build one's own layers on top of that. Unfortunately none of the existing languages are designed in that manner. Brand X is an "assembly language" for designing higher level special languages. The decision to implement one's own customized language is also influenced by the detail to which the representation really "represents" the knowledge. For example, in the context of this project, how much does the system really "know" about the protocol, the disease, the world in general? One finds that there are essentially two kinds of representations and representation-languages.

Those dealing with specific domains so that they have used an abstraction of the problem domain which is sufficient to solve the problem. Examples could be SHRDLU and EL ([Winograd72], [Sussman80])

Those which attempt to deal with more general domains such as some subset of the real world, examples can be FRL, XLMS, KLONE ([Minsky75], [Hawkinson80], [Brachman79]).

It is clear that the evaluation of a Representation Language depends on the context it is being used in and the purpose for which it is being used. We might want to evaluate it with respect to an engineering criteria where one might judge the adequacy of the representation language with respect to an actual knoweledge based system in a particular domain, or one may evaluate it as a potential general purpose language, applying criteria such as adequacy, ease of expression and power with respect to representing knowledge about general domains.

To focus the discussion we regard a knowledge based system to be an interpreter which interprets a data base. For the time being we ignore issues of declarative and procedural styles of representation.

A representation may meet different needs. We examine in turn the relationship of these different needs to the nature and the depth of representation. We consider the case when the representation is :

Just sufficient for solving the problem. If one wants to solve the traveling salesman problem it is enough to consider the nodes of a graph to represent the cities and links with between them to represent the distance, thereby abstracting out other details of the world. Similarly to solve circuits one may represent only as much detail as is necessary using a sufficiently abstract topological description to solve the problem. Just the solution itself is enough. One can consider a program which solves the Rubic's cube problem and "just works". All the semantics and description is in the mind of the user.

Is flexible in the sense of being extensible. This means that there are enough categorical representations on top of the basic system that adding one more of a kind doesn't involve redoing the whole system. Examples can be a program which helps manage inventory control and has enough flexibility in the representaion that a different business does not require changing its fundamental structure.

Is general in the sense of allowing expression of other domains. This means that there is nothing domain specific which prevents the system from being used for other domains *and* that there are general useful features which are applicable to other domains (such as a TMS kind of facility). This essentially means a factoring of the domain specific knowledge and the representation structure so that ot can be adapted to other domains. An example can be EMYCIN.

Is sufficient for explaining the "how" of processes (goal oriented). Sufficiency for describing the "how" of system processes must come from the goals of the system and SHRDLU [Winograd72] demonstrated such elementary capability in terms of keeping a "trace" of events. A more comprehensive discussion can be found in [Swartout80] where he discusses how by automatically generating the explanation from the program the program can be its own representation of what it does. This doesn't necessarily mean a deep semantic description, just that it is a consistent way of building in the explanation of "how" of processes in the program structure.

Is sufficient for explaining the what and why of things and processes. This assumes that there is a description in the representation which allow questions of the form "what is a ...." and "why ..." in a nontrivial fashion in the sense of not being "canned" replies but replies configured from its description of the world (if only in having a deep enough semantic net to begin with).

We now discuss the above needs of representation in terms of layers of descriptions, the minimal ones of which are sufficient to solve the problem but the additional ones help to meet the other needs.

7.6 The bottom up approach rather than the top down one

The additional of a few layers can add to the flexibility or add a semantic description depending on the nature and number of the layers of description. For example on top of the minimal abstract description which solves the problem one can add a few layers which "map" other descriptions into the abstract one. Also within a given layer of description one can break up a lot of structure at one level into many more to make the structure more fine not unlike when one writes subprocedures to break up the procedural knowledge into sharable units or layers. On the other hand one can build layers in the sense of an additional "semantic" description (where there is additional of knowledge of what the symbols stand for, as well as structure, using declarative structures (a semantic net perhaps) to "describe" the relation of concepts at this level to a higher

(in a semantic sense, such as more general concepts) level and to the rest of the world. It is my claim that just adding one or two layers can only add to the flexibility and not the real "knowledge" (in terms of what the systems can "say" about what the symbols stand for in the system, and whether the system has access to its own structure through a declarative description of it). After all what can the system tell us about "what is a protocol" when it doesn't know about the people the protocol applies to , a concept of disease, of health? In other words the claim is that a description which can describe the "what" and "why" questions would require not 5 layers but a reasonably complex ontology of the things in the world in terms of which questions can be answered. For all other purposes one must be content with a "bottom up approach" to include as many layers as are required by the considerations mentioned above. For a reasonably complete description (which will have a deep knowledge of what the symbols stand for) one has to have a "top down approach" beginning with "things in the world" so that there aren't merely "living things, concepts and protocols' in one's universe!.

7.7 The meaning of something is in what it does

Or when to use the procedural or the declarative approach. As discussed above, more layers of description make the representation more structured cleaner and general but one must use some criteria to decide which ones to represent in what detail. Representing knowledge procedurally certainly hides the semantics away in the code but it is sufficient to have some things work rather than have a description of them also. It is enough to have a procedure that checks if a continuous parameter has been represented in all its possible ranges by performing simple arithmetic tests rather than have a more detailed description of what "a complete range of values mean". Therefore at some levels one has to ground the meanings of processes in "what the procedures do" and the meanings of concepts in "what the symbolic strings represent in the mind of the user". However one can use a mix of the two approaches to index even the procedural knowledge so that it can be shared easily. The depth to which descriptions must be represented with enough declarative structure depends essentially on the importance of the knowledge, how often it might need to be modified and the kinds of questions the system might be expected to answer about it or in other words the kind of "knowledge" the system is required to have about the specific piece of knowledge and about itself.

7.8 The layers of representation used in this project

In view of the different needs of representation discussed above the representation used is sufficient for solving the problem in the domain of cancer and more than that it is flexible and extensible. It doesn't have deep semantic layers of "what a protocol is "in the world but it has knowledge of types of protocols and parts of protoco. It is a specific control structure designed to implement a mostly data driven system. In the data base of there is sufficient amount of declarative structure to make it flexible but not enough layers to make explanation possible. The interpreter is basically an instantiator which creates an instance of a new protocol based on the declarative representation of the structure of the protocol and uses constraint propagation through deamons to make its own deductions. The constraints are used to implement the "knowledge". More often than not the constraints are procedural, though structured so that it is easy to add instances of constraint in a certain category e.g. relationships of certain variables to each other.

One major deficiency of the system is its inability to represent processes in a general way. Such a need arises for instance when one is specifying actions for different kinds of response e.g. a relapse is different from the first occurance of the disease and a second relapse is different from a first one. Also to monitor toxicities one needs to observe parameters and again one needs the concept of a process to capture events which are not simple i.e. they can be oscillatory (blood count going up or down) or just repetitive to a finite count. The concept of a compound event (a conjunction of events) helps, which is what exists now. To be really able to describe processes one needs to order the events and bunch them together. A concept of disease progression also requires describing it in terms of a process. For example a process is needed to represent the fact that after a patient has gone through progressive disease for a long enough time there is no possibility of a complete recovery.

The drawbacks in the framework for representation are less serious than the lack of knowledge for the framework that exists already. For instance although there is a framework for representing the toxicities in considerable detail (such as their onset and duration characteristics) it is not always that enough data exists to make use of such a framework or to base a sophisticated algorithm on such a detailed representation.

7.9 Naming

Naming is an important consideration for the processes of both making names meaningful and using the naming process itself to organize knowledge -- one of the motivations behind Brand X.

A name is a unique identifier for an object. Unless there is a way to compose and decompose names efficiently and unless the storage or the indexing of objects is connected directly with names, naming will not significantly affect the assertion or retrieval of information. For instance in bare LISP there is no easy way to reference objects by names. This is because the structure of the object does not play an important role in the naming of the overall object and so the names are arbitrary and to specify them requires the knowledge of the the name itself. However in a language like Brand X where expressions themselves can be unique i.e. serve as names, there are new possibilities to have a more intelligent way to name objects. The advantage is that objects can be their own names and since one can construct them from their parts or decompose them into their parts easily, the process of just specifying their names (i.e. in Brand X to specify the object itself) serves to assert or reference objects. Further if the objects are indexed using syntactic properties then one can exploit those properties to enable faster look-up. The key is to use the structure of the object to specify the object and use that specification to index the object so that it can be referenced more efficiently by knowledge of its parts and (assuming the name or the structure reflects its context) by knowledge of its context.

For example if we are dealing with drugs, one way to name different drugs is to name them [drug 1], [drug 2] etc. Another is to name them as [drug 1 [treamtment 1]] or [[treatment 1] drug 1]. The obvious advantage in the latter case is that some of the context is stored as part of the name and apart from making it more meaningful, it helps in retrieving information which is only partially specified. For instance if we wanted to know whether there is a ball of color red we could see if a structure called [ball red] existed or not rather than exhaustively check all balls or all objects of color red. Of course this assumes that we follow a convention in naming objects so that we can look for them in a systematic way. In Brand X objects have their Car-1 property stored so that it is useful to have the context as the the trailing part of a name so that (for example) by asking which are the objects whuch have the specified object as their car, we can find where this object appears. This brings us to one of the problems of such indexing schemes. Since by choosing to index the objects in one particular way, one imposes a particular view point (such as the first element is the key by which to index the object) the retrieval process is efficient only if a

similar viewpoint is used in the search.

7.10 Future Directions

In retrospect and from ideas which emerged during the course of the project and could not be followed through, there appear to be a number of directions to be persued further in this project.

7.10.1 A Detailed Simulation Of Drug Interaction And Toxic Processes

Although we have discussed earlier the difficulties of a complete full scale simulation of disease/toxicity processes, some incomplete models can still be of some use. The process of toxicity is much better understood than the therapeutic process and much more data is available in that aspect of the effects of the drugs. It would then seem plausible that one can set up first order models to simulate the toxicity processes. We have taken a step in that direction by having a simplified simulation (a zeroeth order one) of the onset and duration of the toxicities, to detect simultaneous occurance of toxicities which might have gone unnoticed otherwise. A more sophisticated system could take into account the dose specific effects of the toxicities, the interaction between different drugs, and the metabolic details of the drug toxicities. In this way one could figure out a more "optimal" way of drug administration, which minimizes toxicity, if it doesn't seriously interfere with the therapeutic process. At least it could take care of at least one part of optimality -- not to kill the patient due to the drugs themselves - analogous to the effort in the digitalis project. The main difficulties seem to be in getting hold of the detailed data about the toxicities.

7.10.2 Violations of the protocol : A structured approach

It would be nice to have a hierarchy of rules which govern the execution of the protocol. For instance protocols frequently mention "Contact the study chairman for instructions" as a response. Less extreme examples are when a certain rule forbids the application of some other rule for a certain period of time. (Dose modification for a certain toxicity may be delayed for intensive induction therapy). When such violations are recorded they could take into account which rules "allow" the violation. Typically higher level rationale will override simple rules which govern most of the simple cases. In the absence of a complete model of the disease/treatment process, this is the only way to encode the importance of one kind of knowledge over another, so that some violations are more important than others. One can think of violations as

"contradictions" in the TMS sense and then "resolve" the contradiction by "retracting" appropriate assumptions or rules

7.10.3 Library of earlier therapies and pattern matching

Much of the knowledge relevant to the design of new therapy is contained in previous protocols. Since all possible effects of different combinations of drugs/treatments are not predictable, one has to usually begin where other studies have left off. It would then be useful to have a library of previous protocols where the details of the treatment and it's effects could be kept in a summarised form so that the protocol designer can browse through earlier protocols and even modify some of them to create new ones. Some simple form of pattern matching could be implemented to retrieve related protocols.

7.10.4 The Logical and statistical design of the protocol

Earlier in the thesis we have formulated the statistical problem. The simple yet useful approach that we have implemented, although being powerful for its size doesn't deal with some of the more involved aspects of therapy design. For instance it deals with the problem of statistical significance as an abstraction without taking into account the specific structure of the design. Understanding of the detailed design of the therapy might indicate that some of the statistical assumptions such as independence of different factors aren't being satisfied and and the program could do some kind of simple pattern matching to isolate such flaws. Also it can suggest a reorganization of the design (in terms of having more/less arms, grouping, sequencing of the treatment) for a better logical design.

A program which knows about the statistics involved and has some knowledge about the logical structure of the treatment could add to the usefulness in the preliminary exploratory design of the trial. The program could understand the structure of the treatment part of the protocol and use the further information supplied about the estimates of the parameters and their spreads to give a really informed advice about the possible achievable levels of significance.

Another possibility is to add some heuristics which can help it choose from among the different statistical interpretations which are available on the basis of the particular population, the epidemiology, the strata in which the patients are divided and the particular disease model. A knowledge of the demography of the region could help it to guess if the needed number of

patients qould actually accrue in the given time periods available.

7.11 Summary

In this final chapter we have examined the relationship of the scope of this project to the success of a clinical trial, the different issues relating to the depth of representation used in the project, touched on issues concerning naming and indexing of objects and pointed out some future directions.

We found that while a more complete assistant to protocol design would require deeper knowledge of processes of drug and disease, assuring the consistency and completeness can be of significant help in the success of a clinical trial, both from the point of view of the patient and from the point of view of the study.

We discussed the different needs that different layers of representation meet and claim that number and the nature of different layers of representation required differ sharply with the needs and the most deep description is required to answer the "what is .." and "why" questions.

We examined the role of naming in the indexing of objects and explored the ability of Brand X in allowing unique reconstruction of objects by specifying their parts, as one way to have a more intelligent system of indexing objects.

Future directions to this project can contribute most easily to making the statistical advisor more intelligent. More significant contributions could be to include a more sophisticated simulation of different processes and a hierarchical system to represent constraints so that violations can be handled in a structured way.

7.12 Conclusion

We have described the background, motivation and the details of implementation of an expert system to help doctors design better cancer protocols. We found lack of knowledge of cancer to be a major limiting factor in limiting the intelligence of this program. We have used our own representation language, built on top of Brand X, and found that to be a good decision, in the context of the relatively abstract description of the domain that we use. While the solutions to some of the computational problems (such as those regarding change) are visible, a significant

growth in the medical intelligence in the program will have to come from knowledge of mechanisms of drugs and toxic processes. The statistical intelligence is the most easily accessible to extension. The user engineering part if the project has been essential in making this program a usable one.

8. References

[Bernstein78]
Bernstein, D. and Lagakos, S.W., "Sample Size and Power Determination for Stratified Clinical Trials", *J. Stats. Comp. Simul.*, Vol 8, pp 65-73, (1978).

[Brachman79]
Brachmann, R. J.,"An introduction to KL-ONE", in Brachman, R. J., *et el*, *Research in Natural language understanding*, Annual report pp 13 - 46, Bolt Beranek and Newman Inc., Cambridge, Mass., (1979)

[Bosyj76]
Bosyj, Michael, "A Program for the Design of Procurement Systems." MIT/LCS/TR160 (1976).

[Carter81]
Carter S.K.*et al* "Principles of Cancer Treatment" (1981)

[Doyle79]
Doyle, Jon, "A Truth Maintainance System", A.I. Memo 521 (1979)

[Hawkinson75]
Hawkinson, L. B., "The representation of concepts in OWL",*Proceedings of IJCAI75*, MIT AI Lab., (1975).

[Hawkinson80]
Hawkinson, L. B., "XLMS: A linguistic memory system", Technical Memo, MIT/LCS/TM-173 (1980).

[Jackson80]
Jackson, R.C., "Kinetic Simulation of Anti-cancer Drug Interactions", *Int. J. Bio-Med Computing*,11, pp 197-224, (1980).

[Martin78]
Martin, William A.,"Description and the specialization of concepts" MIT/LCS/TM-101 (1978)

[Martin80]
Martin William A., 6.863 course notes. (1980)

[Minsky75]
Minsky, M., "A Framework For Representing Knowledge", in *The Psychology of Computer Vision* ed. by P. H. Winston, McGraw-Hill, New York (1975)

[Winkler75]
Winkler R.L. and Hays, W.L., "Statistics", Holt,Rinehart and Winston (1975).

[Peto76]
Peto R., *et el*"Design and Analysis of Randomized Clinical Trials Requiring Prolonged Observation of Each Patient." part I and II, (British Journal of Cancer) **34**, 585 (1976).

[Russ80]
Russ, Thomas A.,"Design of Drug Administration Protocols for Cancer Therapy", Bachelor's thesis MIT/EECS (1980)

[Shortliffe81]
Shortliffe, E.H. *et al*, "ONCOCIN: An Expert System for Oncology Protocol Management", Proceedings IJCAI-81, Vol II, (1981)

[Sussman80]
Sussman, G. J., and Steele, G. L., "Constraints - A langugae for expressing almost hierarchical descriptions", *Artificial Intelligence* **14** (1980), pp 1- 39.

[Szolovits79]
Szolovits, Peter, "Artificial Intelligence and Clinical Problem Solving" MIT/LCS/TM-140 (1979)

[Szolovits80]
Szolovits *et al* "Brand X Manual" MIT/LCS/TM-186 (1980)

[Valeriote79]
Valeriote, F.A.,"The Use of cell kinetics in the development of drug combinations",*Pharmac. Ther.* Vol. 4,pp 1-33, Pergamon press (1979).

[Winograd72]
Winograd, Terry, "Understanding natural langugae", Phd thesis, Academic press, 1972.