A Decision Analytic Approach to Web-Based Clinician Training

by

Lincoln J. Chandler

B.S., Computer Information Sciences
Florida A & M University, 1999

Submitted to the Sloan School of Management
in Partial Fulfillment of the Requirements for the Degree of

S.M. in Operations Research
at the
Massachusetts Institute of Technology

February 2005

Signature of Author: [illegible]
Operations Research Center
January 14, 2005

Certified by: [illegible]
Richard C. Larson
Professor of Civil Engineering and Engineering Systems Division
Thesis Supervisor

Accepted by: [illegible]
James B. Orlin
Edward Pennell Brooks Professor of Operations Research
Co – Director, Operations Research Center

A Decision Analytic Approach to Web-Based Clinician Training

by

Lincoln J. Chandler

Submitted to the Sloan School of Management
on January 14, 2005 in Partial Fulfillment of the
Requirements for the Degree of SM in
Operations Research

ABSTRACT

Given the desire to create interactive websites that effectively engage and instruct medical professionals, an alternative model for online case studies was developed. The resulting application presents the user with a virtual patient, asks for information regarding the treatment and care of that patient, and provides customized feedback to the user. When a person uses this application, one could say the goal of the user is to make the necessary decisions that will stabilize the patient, and the goal of the application is to provide feedback regarding those decisions.

In order to adapt to user decisions, the design incorporates an unconventional use of decision analysis. The source of uncertainty is the clinician's strategy, or sequence of decisions. Given the user's decision, the appropriate system response is assumed to be uncertain *a priori.* The proposed model requires the application to conduct an internal analysis, and then condition the response on the circumstances under which the decision is made. This conditioning approach informs the patient's behavior during the simulation, and it determines the appropriate constructive feedback for the user. Intuitively, a system constructed using the proposed model is better suited to address the educational needs of an individual learner. Also, despite the context of this model, it is noted that the proposed model need not be restricted to medical applications.

Thesis Supervisor: Richard C. Larson
Title: Professor of Civil Engineering and Engineering Systems Division

(This page is left intentionally blank)

ACKNOWLEDGEMENTS

Primary funding for this project came from a joint initiative between the Massachusetts Institute of Technology and the Brigham and Women's Hospital; principal investigators were Dr. Richard Larson (MIT) and Dr. Robert Rubin (Brigham). Ms. Melinda Cerny served as the project manager. Supplemental funding was also provided via a student fellowship award from the National Science Foundation. Technical advice on writing code for the prototype was provided by Dr. David Craft (ORC) and Mr. Eric Brittain (EECS), both of MIT.

Foremost, I give thanks to God and to my family: Raymond D., Audrey P., and Simon J., your love and your support are my foundation in all that I do. Even when I feel weak, I can continue because your faith in me is strong.

To my thesis advisor, Richard Larson: I can not thank you enough for your guidance throughout this process. I appreciate your dedication to helping me define and achieve my goals. Your suggestions and your willingness to be candid with me at all times attest to your genuine interest in my success. As my journey continues at MIT and beyond, I will strive to improve my command of the many lessons you've taught me.

To my project directors, Melinda Cerny and Bob Rubin: You provided the setting for this research, and your guidance and perspective were invaluable. I sincerely hope that this research is the step in the direction of defining the next generation of accessible, effective online education. It was truly an honor to work with and get to know you both.

To the case author, Dr. Sigal Yawetz: I am very grateful for your assistance in the implementation phase of this project; your contribution helped convert these concepts into a working reality. I am aware of the very hectic circumstances that you had to work under over the course of this project, and I really appreciate your dedication and commitment to this project.

To my department, the Operations Research Center: I could not have asked for a better group of friends and colleagues. From the administrators to the students, you all have collectively been so generous with your advice and your time. Special thanks to Jim, Paulette, and Laura, and my friends from the entering class of 2001.

To my extended MIT family: The deans and staff of the Graduate Students Office provided my introduction to MIT and they have welcomed me with open arms ever since. Some of my strongest friendships have been formed spending time with my brothers and sisters of the BGSA and the BSU; thanks for providing a cultural home away from my HBCU roots. Last but not least, I must thank those special friends at MIT who strive to reach the ACME of their respective disciplines through dedication to accountability and getting goals accomplished.

(This page is left intentionally blank)

TABLE OF CONTENTS

(This page is left intentionally blank)

CHAPTER 1: INTRODUCTION

Motivation: Online Case Studies

Our discussion begins with a consideration of educational computer programs that support interactive, online case studies. In the medical community, these modules are used to illustrate issues of medical practice and theory to novice clinicians. Case studies are a popular teaching method in medicine and other disciplines because they provide a way for seasoned professionals to convert their actual experience into an educational narrative for the learner. In a case study, the case author presents background information and assessment of a particular patient; the interaction occurs when the learner is required to provide his or her own thoughts regarding care and treatment.

Design of traditional case study modules

From observing a number of medical websites that feature case studies, one can conclude that these case study modules are generally developed and presented in a deterministic, or fixed, manner. Relevant information about the patient (e.g. vitals, X-Rays, lab results, etc.) is provided in a specific order, at predetermined times throughout the case. Also, the user is generally not required to request access to this data; it simply appears when the user reaches a certain part of the case.

When the user submits a diagnosis or chooses a therapy, these modules associate a single outcome with that decision and that outcome will never change. Also, with very few exceptions, the design of these modules will only acknowledge one correct choice for any given situation. This notion of a single correct decision is enforced explicitly; if an improper decision is made, the user is immediately notified of the mistake and is either given the correct answer or instructed to go back and try again. In either case, the arc of the case study operates along a single path that the user must be guided back to before another decision can be made.

This "single path" structure is depicted in **Figure 1**; note that every potential decision has only one possible outcome. For the first decision, the depiction shows that regardless of the user's choice, the next system state is fixed. For the second decision, the user cannot move forward until a particular option (i.e. the "correct" answer) is chosen. This methodology implies that all users who navigate the course will be given a common set of decisions to make and, when the case is over, that all users will be lead to a common conclusion.

The main advantage of this approach is the simplicity of implementation. Because all users will have essentially the same experience, there is no need to keep record of an individual user's past decisions. The online case module, in its entirety, is typically little more than a collection of static HTML pages; the information used to link the pages is explicitly coded within the pages. Each choice has a consistent, predetermined outcome, so once the links have been established, no further computation by the system is necessary.

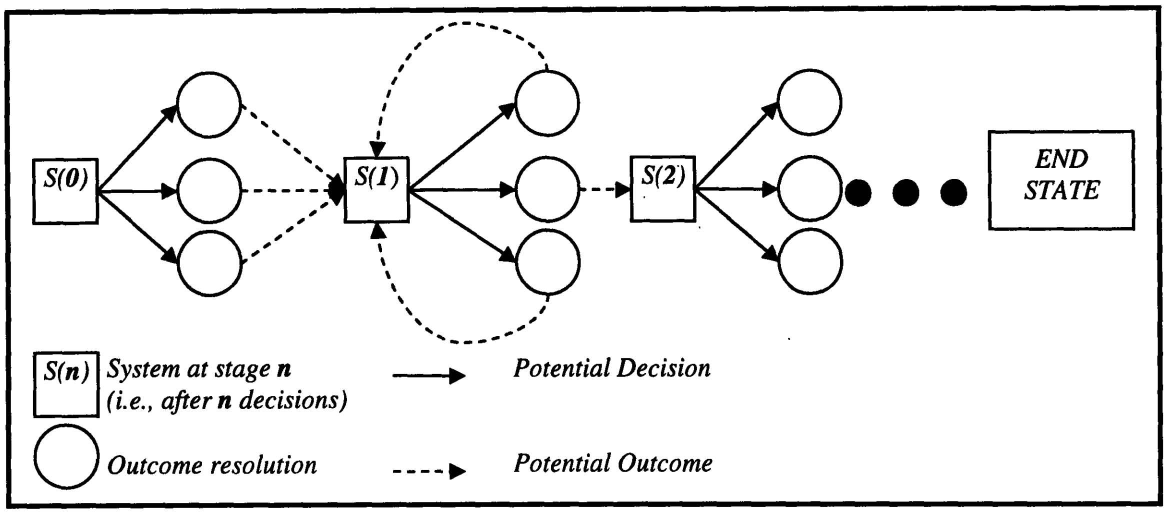


Figure 1 - Depiction of traditional case study structure

The user experience

When an online case study begins, the user is provided with a narrative describing the patient's symptoms, and a set of supporting visuals, such as photographs, slides, etc. Over the duration of the case, the user is presented with a series of decisions to be made. These decisions are typically represented as multiple-choice questions that concern the care of the virtual patient and/or general knowledge of the scenario presented. After each decision, the user is presented with feedback regarding their choice, and any other comments regarding the decision. Also, updated statistical and/or physical information is provided to help guide the user deeper into the case.

For the user, the current system typifies the trade-off between simplicity and realism. In its current design, a user can navigate through a case relatively quickly and gain some level of insight regarding the care of a patient. Given a patient and his symptoms, the case study conclusion will reveal the treatment that gives the patient the best odds of recovery, as well as the recommended tests to perform along the way. This information is a key component of a clinician's expertise, but the case studies rarely mention potential side effects or efficacy rates, which are crucial to consider in a field of very few absolutes. In practice, the lack of certainty in medicine can sometimes lead to competing notions of a patient's care. In contrast, the case study modules in use can only support a single treatment strategy, so alternative therapies are usually mentioned only in passing, if at all.

This is not the only instance where realism lags behind simplicity. Another trade-off can be seen in the way important test results and visual records are passively presented to the user. In a real clinic, a clinician would need to decide what tests to run to evaluate the patient and a clinician would also need to interpret the data received, i.e., figure out the relative importance of the available metrics.

Limitations of the Existing Model

Although the traditional case study design is well-received by the medical community, its static structure limits its effectiveness as an engaging, reusable online learning experience. In seeking to improve the experience, there is an assumed correlation between the level of realism and the ability to engage the learner. Suggestions for improvement are then made under the following premises:

1. Clinicians could receive better training in a simulation style environment that offers a more realistic model of patient-clinician interaction.

2. The introduction of probabilistic outcomes and alternative scenarios would naturally expand the scope of any given case study; extending this idea, by developing a model for navigating a stochastic case environment, the foundation is set for a learning environment with a broader scope, increasing the potential for an engaging experience.

The case for improved realism

As noted earlier, case study modules tend to follow a "single path" methodology, so when a decision is made, there is no variability in the outcome. The lack of variability limits the scope of the case; once a user navigates a case successfully, the value of reuse (that is, the added educational benefit of revisiting the case again) is low. Also note that, in the majority of cases, there is a single strategy, or sequence of decisions, that represents the "correct" way to navigate the case. When the user deviates from this sequence by making an "incorrect" decision, the case study module is designed to either:

- provide a description of why the choice was incorrect and guide the user back, or;

- present the correct decision with justification followed by <u>all</u> of the incorrect decisions and reasons why they are incorrect and move the user along.

In either case, the learner is never given accountability for making an improper decision. In the latter case, this limitation is compounded by a standardized mechanism for feedback. As a result, the user's insights are limited to following the case author's approach, rather than developing the broader perspectives associated with experience – based learning.

By creating a more user-dependent educational experience, one can facilitate a more realistic mode of learning. Although the same body of information is passed to the learner, the order of presentation is conditioned on the user's actions. This property is especially powerful when a learner is performing poorly; by allowing the user to stray from the "best path", a more realistic model challenges the user to address and understand the consequences of their specific course of action. Also, a model that permits the user to make mistakes can use this information constructively to provide comments and feedback specific to individual performance.

The case for variation and alternative scenarios

The inherent uncertainty in clinical practice requires added care when providing feedback. Clinicians using a case study module should receive feedback based on the decisions they make, not on the outcomes of those decisions. Unlike the existing case study models, in which case information and feedback are delivered together, an alternative model may seek to decouple the analysis of the user's decisions from the presentation of outcomes in the case. One approach would be to delay feedback until the simulation has ended. From an educational perspective, this method provides a means to counter the perception that positive results imply sound technique, and vice versa.

In addition to the individual benefits of adopting a more user-driven approach, there are long-term benefits to consider as well. The lack of variation in the prevailing case study model implies that the recommendation of a single "best path" is frozen in time. If a better approach to treating the patient were discovered, the recommendations (and therefore, the case) would become obsolete. In a more realistic model, a new treatment approach would equate to adding a new path and perhaps updating the feedback, but the functionality of the rest of the model would be unchanged.

Overview of Proposed Model

To address the shortcomings of the traditional online case study, the alternative model proposed in this paper has a dual functionality. The primary function is to support an interactive self-assessment of the user's medical knowledge in a given area, and the secondary function is to simulate a patient care experience. The proposed model achieves these means by defining a system that initially behaves as a simulation would; while in this mode, the patient's response to treatment is the primary source of information for the user. This mode of operation is allowed to continue only if the simulated scenario is of educational value and if the user is exercising reasonable judgment in caring for the patient.

If either of these conditions is violated, the system is defined to be in a "pruning state". The term "pruning state" is motivated by the notion that the model design should circumvent undesired states of the system as one would prune unwanted branches of a tree. Once the system enters a pruning state, the current simulation is halted and the user is presented with feedback on their performance. This feedback includes a description of the event that caused the pruning in addition to an analysis of the user's overall strategy; i.e., all of the decisions made by the user. The user is then prompted to backtrack and retry a previous decision, restart the simulation, or exit the system.

It is important to note that the patient and case updates (presented to the user per iteration) are generated by the outcome. This is contrast to the user feedback, which is based on the conditions under which the decision is made, and is determined earlier, before the outcome is known. Although generated during the simulation, feedback on a user's strategy is not shown until the simulation is halted. This approach allows for a learning tool that separates the analysis of the user's decisions from the simulated effects of those decisions.

Review of Literature

Online Medical Education and Case Studies

Like practitioners in other fields, members of the medical community have been eager to embrace new opportunities for education via the Internet. A prominent indicator of this interest is the growth in the number of websites that are classified as examples of online continuing medical education, or online CME. A practicing physician is required to stay abreast of advances in medicine; the formal proof of this activity is the accrual of CME credits from certified providers[1]. Traditionally restricted to coursework and computer programs on CDs, the Internet has afforded the opportunity to build and distribute CME materials online. As evidence of the explosive growth in online CME, a study conducted by Sklar in 2000**[4]** identified a total of ninety-six websites offering over 3,000 hours of CME credit; by Fall 2004, Sklar's index of online CME

[1] The Accreditation Council for Continuing Medical Education (ACCME®) is the governing body for this activity; more about ACCME can be found online at http://www.accme.org

resources had identified over 280 sites, offering more than 13,000 courses and over 23,000 hours of CME credit.[2]

Case studies, which are an established element of classroom education, have become a popular feature on several online CME sites as well. The notion of a "case study" can vary depending on the context; for this discussion, a suitable characterization defines a case study as an "active learning strategy depicting a real-life event or situation that is relevant to the professional activities of learners."**[1]** Case studies promote the use of learning from past observation; in a conventional implementation, the elements of the case are examined via group discussion, with an instructor as moderator. The discussions serve to promote critical analysis from the learner, and to this end, the most effective cases tend to involve ambiguous situations with no "obvious" course of resolution. Given the complexities of providing medical care, it should not be surprising that case studies are a well- established tool of clinical training.

Uncertainty and Decision Analysis in Medical Research

There is a sizable body of research devoted to understanding uncertainty in the medical realm, ultimately with the purpose of facilitating an analytic approach to patient care and medical training. Drawing from Szolovits **[5]**, one could argue that research on medical uncertainty tends to belong to one of the following four categories:

- Efficacy evaluation of a given treatment methodology;
- Reduction of uncertainty in diagnosis via statistical analysis;
- Discovery and exploration of factors that influence a physician's decision-making; and
- Causal analysis of how patients make decisions about their own care.

The first category addresses a fundamental question; what treatment strategy is best for the patient? Studies of this sort can be addressed quite effectively through the application of Decision Analysis (DA). The value of DA to medical practice was recognized early on; an example application can be found among Raiffa's earliest works on the topic **[3]**. In his example, Raiffa posits the characterization of patient treatment as a multi-attribute decision tree; he also proposes a method of developing a utility function to represent the overall state of the patient.

Extending Raiffa's example, prime application areas for Decision Analysis have included the development of proper care protocols and within the last thirty years, the development of "quality-of-life" metrics for describing the physical and emotional health of a patient. Applications of the latter sort have proven to be somewhat controversial, as they require practitioners to quantify and compare dissimilar and sometime subjective measures, such as functional impairment and patient comfort. **[2]**

When considering the various applications of Decision Analysis research to medicine, it is noted that typically, the immediate goal is to suggest the proper clinical decision, or sequence of decisions, in an uncertain environment, i.e., patient care. In the long-term, these insights would then be used to address the question of what should be taught in terms of clinical best principles. The model presented here also has a similar immediate goal, but the long-term objective is quite different. For this application, it is assumed that there exists a set of clinical principles regarding the care and treatment of a particular patient. Given this body of knowledge, the goal of this research is to create a more engaging and effective way to introduce these principles to a less experienced clinician.

[2] Sklar's index of Online CME sites can be found online at http://www.cmelist.com/list.htm

(This page is left intentionally blank)

CHAPTER 2: PROBLEM ANALYSIS AND FRAMING

This section outlines the design notions that ultimately lead to the proposed alternative, referred to hereafter as the Compressed Decision Tree (CDT) model. The basic inspiration is the alternating decision tree model, which is developed first. For the intended domain, it is noted that the educational content and analysis will generally not be exhaustive, a fact which exposes the limitations of a standard decision tree implementation. Next, the notion of contextual analysis is used to develop a method of overcoming these limitations, first by defining pruning conditions, and then by generating a system response for each user decision as it is made. This process is referred to as a contextual analysis test; these tests are then the basis for a new analytic structure, the compressed decision tree (CDT). The CDT structure is the essential asset, as it enables the model to simulate the clinician-patient experience.

The Alternating Decision Tree

Given an objective and a set of choices, a decision tree is visual tool that is used to assess the relative value, or utility, of any given strategy (i.e., a sequence of one or more decisions). The entity in charge of picking the strategy is the Decision Maker. The validity of a strategy is dependent on the goal of the Decision Maker and the forecasted outcomes of the strategy. Each outcome has a utility with respect to the decision maker's goal; the decision tree then serves as a mechanism for comparing the desirability of decisions based on their expected outcomes. An alternating decision tree reflects, in a more specific sense, the presence of another "decision maker" (a competitor, or the element of chance, for example) with the power to affect the outcome of a strategy.

The Clinician-Patient Experience

The use of an alternating decision tree model is motivated directly by the nature of a clinician – patient interaction; in what follows, the clinician is the Decision Maker. In practice, a clinician begins with some information about an unstable patient, and a set of decisions to make, such as ordering tests or administering some form of treatment. The clinician makes a decision based on the expected outcome, but the actual outcome is uncertain *a priori*. After the decision has been made, an outcome is observed, and the clinician is given new information and a new set of decisions. This begins a cycle of patient care that repeats until the patient stabilizes or dies; this event referred to as an "end state" for this process.

By defining the patient's initial state as the root, the set of possible sequences of decisions and outcomes forms an alternating decision tree, depicted in **Figure 2**. Note that, in this system, choices can have multiple outcomes. The leaves of this tree represent multiple end states, each corresponding to a different path taken by the user. If this entire decision tree was a representation of the set of all possible case outcomes, then a traditional case study model, like the ones described earlier, would only be capable of representing a single path in this tree.

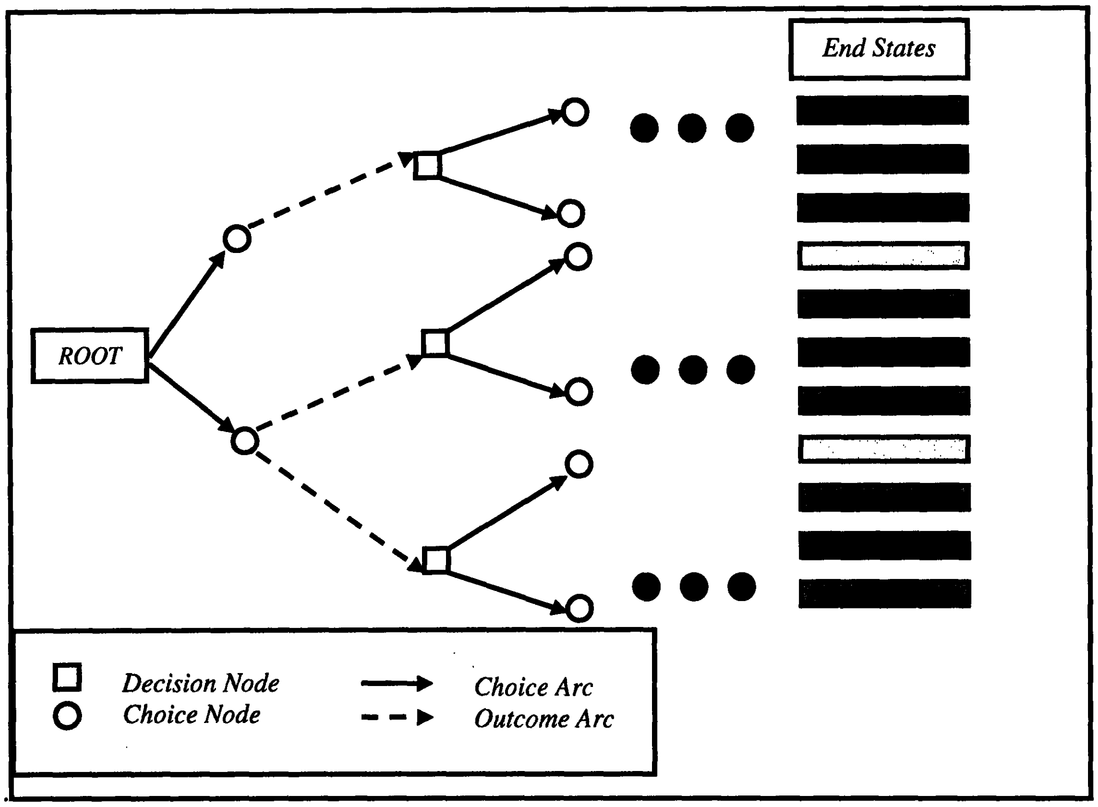


Figure 2 - An Alternating Decision Tree Framework

Additional Needs

An alternating decision tree model may appear to be applicable as is, but proper implementation requires some care. For educational purposes, the model must be able to provide feedback on any strategy (or path) the user is allowed to take. This means that model will require a method of eliminating, or pruning, the paths that the case author will not be able to accurately describe. Also, for users who fail to show basic understanding of how to treat the patient, there is a need to detect poor strategies as early as possible and prune those as well.

There should also be an efficient means to characterize identical sub-trees. Identical sub-trees occur in a decision tree whenever there are at least two decision nodes for which the likelihood of any future event is identical. **Figure 3** illustrates this point using a decision tree that begins with a coin flip; if the next decision is based on whether the decision-maker wins or loses the coin toss, it doesn't matter whether the decision-maker calls heads or tails.

A decision tree model regards each sequence of decisions as a distinct path, but, for a simulation of patient care, it is quite possible that two (or more) distinct paths may be probabilistically identical. It follows that at these points, the model's behavior should be identical. A naïve implementation of a decision tree would generally not take advantage of this symmetry, nevertheless, there is a need to quickly identify sub-trees and handle them in an efficient manner.

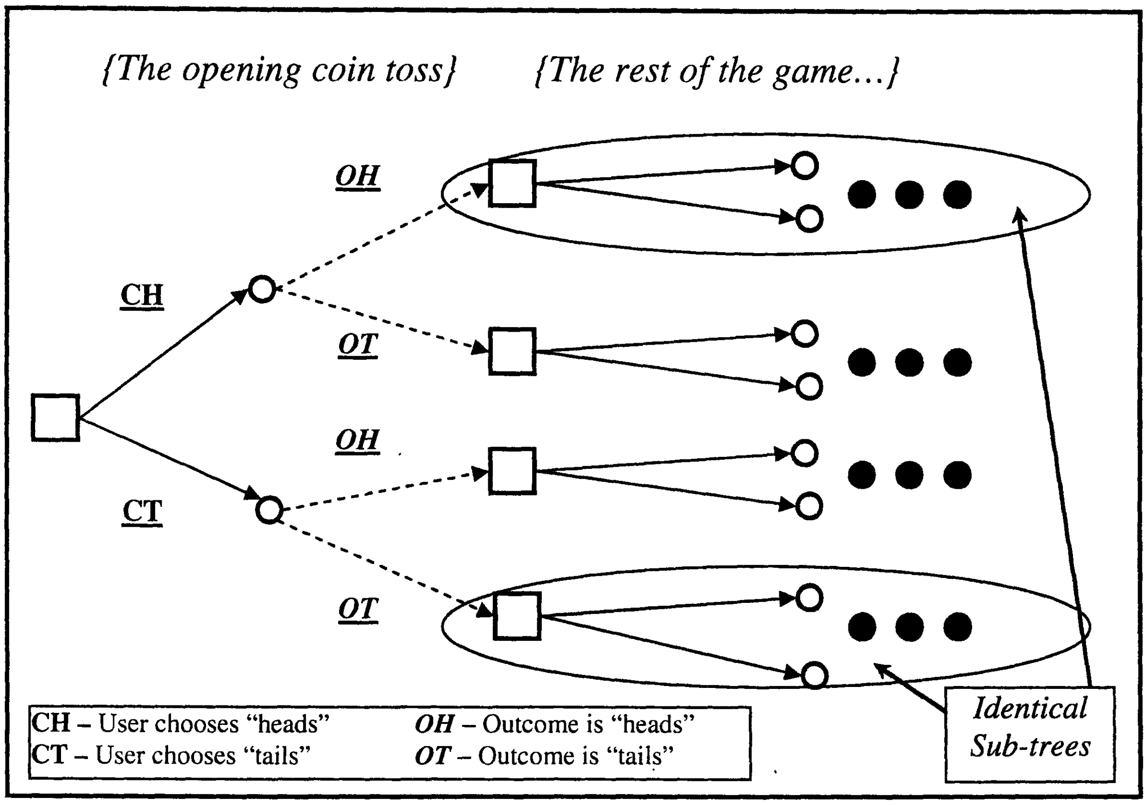


Figure 3 - Alternating Decision Tree for a Football Team

Contextual Analysis

It has been indicated that the alternating decision tree is a reasonable conceptual basis for a new model. However, in order to provide flexible analysis of a clinician's decision-making, the alternating decision tree model requires modification. In what follows, the use of contextual analysis is motivated as a means to provide the necessary enhancements.

In this setting, the term "contextual analysis" is used to describe any process that accepts some request as input, and then gathers information about the state of the process in order to determine the appropriate output. For example, the game "20 Questions" progresses as a contextual analysis would; the initial request is to determine an unknown object, and the player's task is to ask a series of questions about the object with the goal of determining the correct answer. In the world of medicine, contextual analysis appears frequently in the form of treatment flowcharts; for a treatment flowchart, the appropriate treatment regimen (i.e., the output of the analysis) is conditioned on the specifics of the patient's medical condition.

Initial Application: Pruning Conditions

Recall that the scope of a case study decision tree is limited by the amount of information the case author is able to provide. Because of this, it is imperative that the "boundaries" of the simulation space are explicitly defined. This provides an initial use for contextual analysis; if an analysis were to indicate that a boundary has been violated, the simulation would be programmed to stop. This measure ensures that the user will only encounter scenarios that the simulation is programmed to handle, which implies that the simulation will remain pedagogically relevant to the user's education.

In determining the set of viable scenarios, it is not advisable to attempt to enumerate and evaluate all of the possibilities; decision trees grow exponentially, so this approach is not scalable. Taking another view, if given an arbitrary case simulation in progress, the interest lies in determining the general conditions under which the simulation should be stopped. Formally, these are the pruning conditions for the model. In this model, there are four types of pruning conditions defined. If any of these conditions are satisfied, the simulation is said to be in a pruning state. These pruning conditions provide a means to broadly describe the boundaries of a decision tree; the associated pruning states represent the set of simulation scenarios that are "off limits" to users. The pruning conditions are defined in **Table 1**. These conditions are not mutually exclusive, and any one is sufficient.

PRUNING CONDITION	INTERPRETATION
UNDEFINED	1. The program cannot convey or simulate the outcome of the user's decision.
OUT_OF_SCOPE	2. The outcome of the user's decision can be described, however the patient's condition is outside of the educational scope of the module.
USER_ERROR	3. The user has made enough poor decisions to warrant intervention.
SYSTEM_QUIT	4. The user has reached an end state (i.e. the patient is stable or dead), or the user has halted the simulation.

Table 1 - Pruning Conditions for the CDT Model

Applications Beyond Pruning

Contextual analysis can also be used to identify the presence of identical sub-trees in an alternating decision tree. As mentioned earlier, identical sub-trees occur in a decision tree whenever there are at least two decision nodes for which the likelihood of any future event is identical. Plotted on a decision tree, the simulation patient state at a given point is uniquely determined by the sequence of user decisions and patient outcomes that have occurred up to that point.

The decision-outcome sequence provides a sufficient amount of information for the model to simulate a response, however, this amount of detail can be much more than necessary. As an alternative, a direct query of the patient's status is usually a more concise indicator of the patient's probable response. Consequently, a well-formed contextual analysis of the patient state can be equally as informative as the decision- outcome sequence, and typically more efficient.

On a larger scale, contextual analysis can be used to govern the behavior of the entire program. As the simulation iterates, an analysis of the patient's state and the state of the system can be used to determine whether the simulation should continue or halt. The same analysis can also determine more specific behavior; for example, the availability of additional information may be dependent on a measure of elapsed time.

Creating the Compressed Decision Tree

In this section, the Compressed Decision Tree model is introduced. First, a fundamental requirement of the model is addressed; that is, the ability to process a single decision. Ultimately, the further application of contextual analysis is used to compress the functionality of a traditional decision tree into a single-step tree.

The Contextual Analysis Test

When considering an executable prototype of the model, it is reasonable to model the simulation process as a loop; the exiting condition for this loop would be the presence of a pruning condition. The instructions within the loop represent the rules that govern the simulated effects of each user decision. In order to process a single user decision (and simulate an outcome of that decision), There are four criteria identified that need to be addressed; they are presented below, in order:

{C1}	Is the user performing well enough for the program to continue?
{C2}	If so, is it possible for the program to assign the relative likelihood of outcomes?
{C3}	If so, is the (randomly) chosen outcome describable and educationally relevant?
{C4}	If so, should the simulation continue?

If all of these criteria can be met, the program will provide the information necessary to simulate the effects of the decision on the virtual patient, and the simulation will continue. If not, then one of the pruning conditions have been satisfied, and the current simulation will be halted. Together, the four criteria represent a contextual analysis test (CAT) for the submitted decision. Each criterion is addressed by considering a different set of system attributes and, as depicted in **Figure 4,** each criterion plays a specific role in carrying out the simulation. These roles are denoted by *italicized text* in the picture; for example, user feedback (re: the given decision) is determined during analysis of the first criterion.

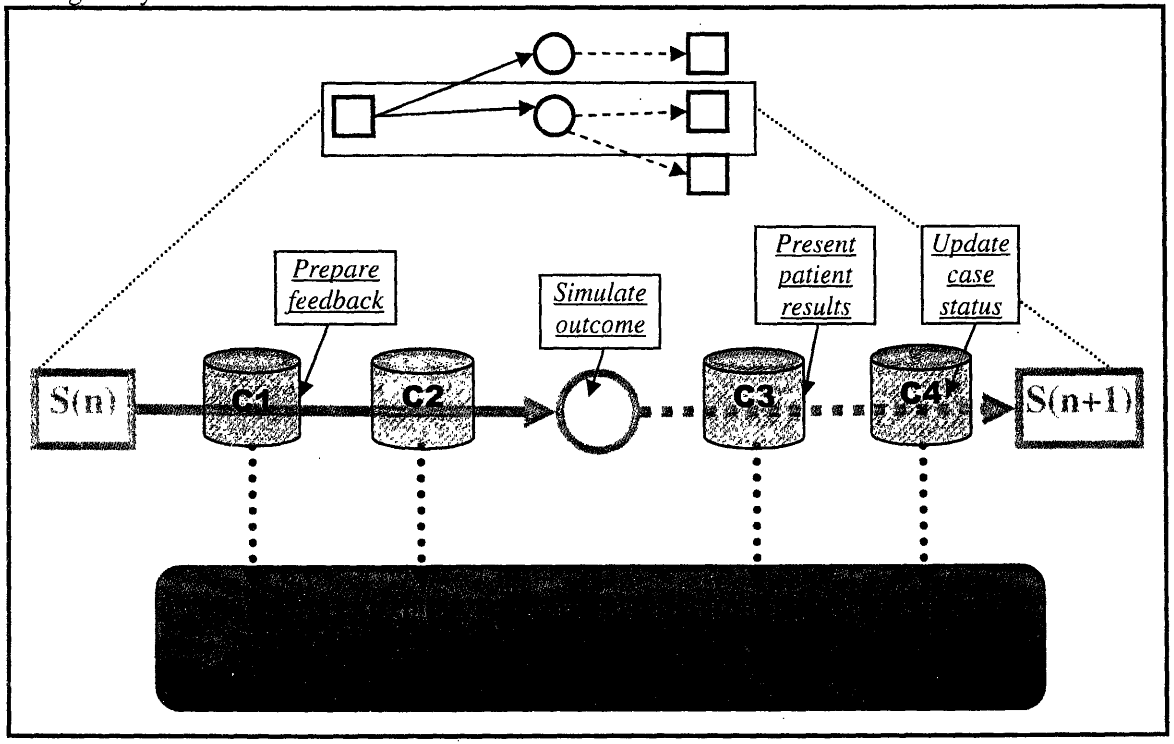


Figure 4 - Illustration of the Contextual Analysis Test

For each criterion, the model requires a minimal amount of information about the system in order to determine the appropriate response. This set of information is defined in the model as a *conditioning event*; this is because this information represents a set of conditions that, if met, are sufficient to determine the behavior of the system and move the program along. The first three criteria are decision-specific, meaning that the conditioning events that are checked depend on the decision that is submitted. Question 4 is global; the conditioning events are the same for every decision.

In this model, all of the information needed to properly simulate a single decision (and its associated outcome) is managed by the decision-specific elements of the CAT. Furthermore, all of the information regarding program behavior is managed by the global element of the CAT. Consequently, there is no longer a need to store an entire decision tree in memory. By defining conditioning events, specifically by using contextual analysis, the physical representation of the tree is compressed into a compact, yet effective form.

The Compressed Decision Tree

If one were to connect a series of these CAT structures for the set of decisions that a user may encounter, the resulting structure would contain enough information to govern the behavior of an entire simulation. This new structure is referred to hereafter as a Compressed Decision Tree.
Figure 5 provides a visual representation of the Compressed Decision Tree (CDT) model. The CDT model has the familiar structure of a traditional decision tree design, with a crucial difference. The depth of this tree is two; the tree need only be deep enough to move the simulation from iteration ***n*** to iteration ***(n +1)***. The elements **D(1), ..., D(d)** represent the decision set for the current iteration.

As shown, the outcome node is surrounded by the CAT elements. Before an outcome is computed, the CAT determines the aptness of the user's strategy and the relative likelihood of outcomes; feedback for the user is collected at this point and held until the simulation ends. After the outcome has been decided, the CAT determines the utility of the outcome and how the simulation should proceed; updated case information, including the next decision set, is determined in this step and presented to the user immediately.

When the program presents the user with a scenario and a series of (decision) choices, it is analogous to posing a multiple-choice question to the user; the set of possible choices is referred to as the decision set. In developing the CAT, it was assumed that the quality of a decision is somehow dependent on the state of the patient, or the system in general. This is generally true whenever the user is required to treat the patient. However, it is noted that in some cases, the outcome of a decision will be the same regardless of the patient or system state.[3] When this occurs, there are fewer conditioning events to consider. In such cases, the analysis is simplified, and only a subset of the criteria listed in the CAT definition is necessary.

Features of the CDT Model

For cases built using the CDT model, recall that the user is not given feedback regarding their decisions until after a simulation has ended. This is a key departure from the traditional case study module; this approach stems from the desire to make the experience more realistic.

[3] For example, consider a general question about the behavior of a given disease. Such a query would have little to do with the particulars of any given simulation; the query is rather a general test of user knowledge.

However, when a simulation has been halted, it was decided early on that the user should be able to view their feedback and retry the simulation without having to start from the beginning. This functionality is referred to here as backtracking.

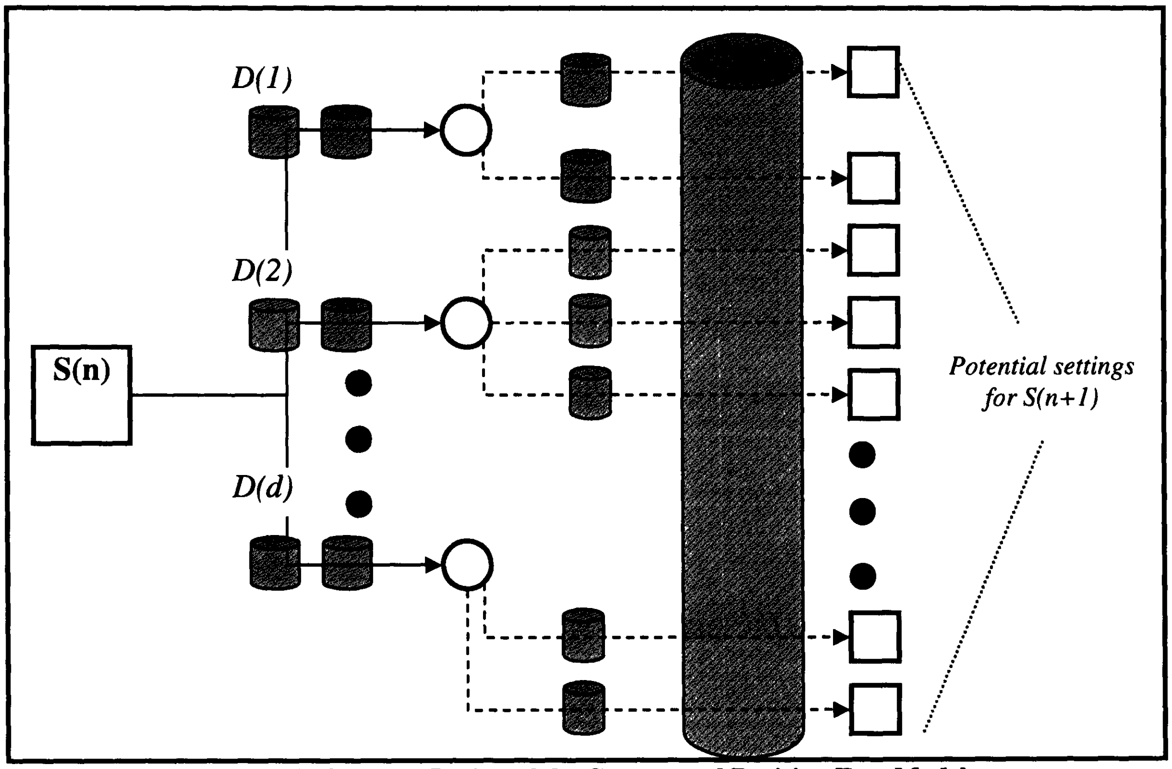


Figure 5 – Physical Design of the Compressed Decision Tree Model

Once a response has been selected, it is checked to see whether pruning is required. If the node does not require pruning, a copy of the system state is stored, and the program continues.

For the CDT model, the process of backtracking is facilitated by storing a copy of the system attributes after each decision is processed. This copy includes the internal patient data, as well as a record of case updates and onscreen information. The copies are stored in a vector, and should a user decide to backtrack, the appropriate system copy is pulled from the vector and all future copies are destroyed. The selected system copy is then used to reset the parameters of the system and the simulation can continue.

It has been shown that the CDT Model provides the means to prune undesirable patient scenarios, recognize identical sub-trees, and reset the simulation to any prior state. This section concludes by noting an additional benefit: the support of multiple patient types. In practice, a clinician's treatment strategy could very well depend on the age, sex, or prior history of a patient. Treatment can also vary due to the progression of illness, presence of allergies, or a host of patient-specific details. A traditional decision tree has a single root, which, in model would equate to having a single patient, at a specific initial state. The advantage of the approach used in the CDT model is that its functionality is determined by the attributes of the patient. Instead of a specific patient as its "root node", the CDT model only assumes a patient in an arbitrary, unstable state. This relaxation allows the model to simulate treatment of the same condition for a variety of patients.

(This page is left intentionally blank)

CHAPTER 3: PROCESS DESIGN

In this chapter, the process flow for the model is defined. Readers should note that this chapter focuses primarily on the mechanics of the CDT model, as opposed to the ideology; accordingly, there is an examination of the primary elements of the model, and how they relate to each other as a simulation progress. Also, there is an overview of the subroutines that drive the execution of a program using the CDT model.

Key Process Elements

The proposed model evolves in successive stages, or iterations. An iteration of the model is defined by the submission of the user's decision, coupled with the effects of that decision. In what follows, let $\{n \geq 1\}$ be the index of an arbitrary iteration of the model. Although the CDT model emulates the behaviors of a decision tree, the structure represents a single-step process that repeats iteratively. Therefore, assuming the simulation has not been halted, it is perhaps helpful to think of the CDT model as a single set of dependent elements with attributes that evolve over the life of the program. **Figure 6** provides a graphical interpretation of this view and serves as the visual template for the process flows that are described later in this chapter. The arrows show how these entities influence one another; the boxes depict the information flows that are determined during execution and sent to the user. One should note immediately the relationship of patient status (Patient(n)) to user error (Err(n)); both entities directly influence the overall system (S(n)), but there is no direct influence on each other. This underscores the aspiration to maintain the distinction between desirable decision-making and desirable outcomes.

The elements shown in **Figure 6** are defined below; although DSet(n) and the HALT subroutine are not shown, their presence should be implied. These definitions are intended to provide a physical interpretation of the elements required for the (active) model and how their values are updated over the lifetime of a simulation. The following section will define a series of processes that permit these elements to interact as intended.

S(n) – State of the model after the n^{th} iteration. The main indicator within S(n) is binary; either the simulation has been halted (HALT), or the simulation is still in progress (ACTIVE). This binary value in S(n) *is determined by* the state of the patient and the severity of user error.*

Patient(n) – State of the patient after the n^{th} iteration of the simulation. This state is representative of the medical condition and history of the patient. The range of possible values grows with the complexity of Patient(n), but the implication is again binary; either the patient's condition is either stable enough to continue the simulation, or prohibitive to going further.

The patient state is conveyed via a combination of the information presented to the user and the underlying conditions that drive the behavior of the simulation. Its value is determined by the outcome of the user's decision and the prior state of the patient, i.e., Patient(n-1).

Err(n) – The severity of the user's tactical error after the nth iteration. This permits the model to keep track of the overall quality of the user's decision-making. When the severity

* Here and elsewhere in this section, the phrase "A *is determined by* B and C" is read as follows: "In all cases, the value of A can and will be determined by some combination of the values of B and C (i.e., B alone, C alone, or B and C together)."

exceeds some threshold, the simulation is to be halted. Although correlation is to be expected, this metric is not determined by the patient state at time ***n***. The value of Err(***n***) is determined by the previous state of the system (S(***n-1***)) and the user's decision.

Outcome(***n***) – This is the probabilistic effect of the n^{th} decision on the patient. This outcome is selected from a set of possible outcomes with a pseudorandom number generator. The relative likelihood of those outcomes (P(n)) is determined by the decision made and the previous state of the patient.

Decision(***n***) – This is the user's n^{th} decision. Once submitted, it drives the aforementioned elements of the model. Any feedback regarding the user's decision is determined by the previous state of the patient and delivered after the simulation has been halted.

DSet(***n***) - The set of decisions available to the user for Decision(***n***). DSet(***n***) is determined by the previous state of the system, S(***n-1***).

HALT subroutine – Entered only when a simulation has been halted. In this subroutine, all feedback regarding the user's performance is presented. The user then has the option of restarting the simulation from the beginning, or from any earlier iteration.

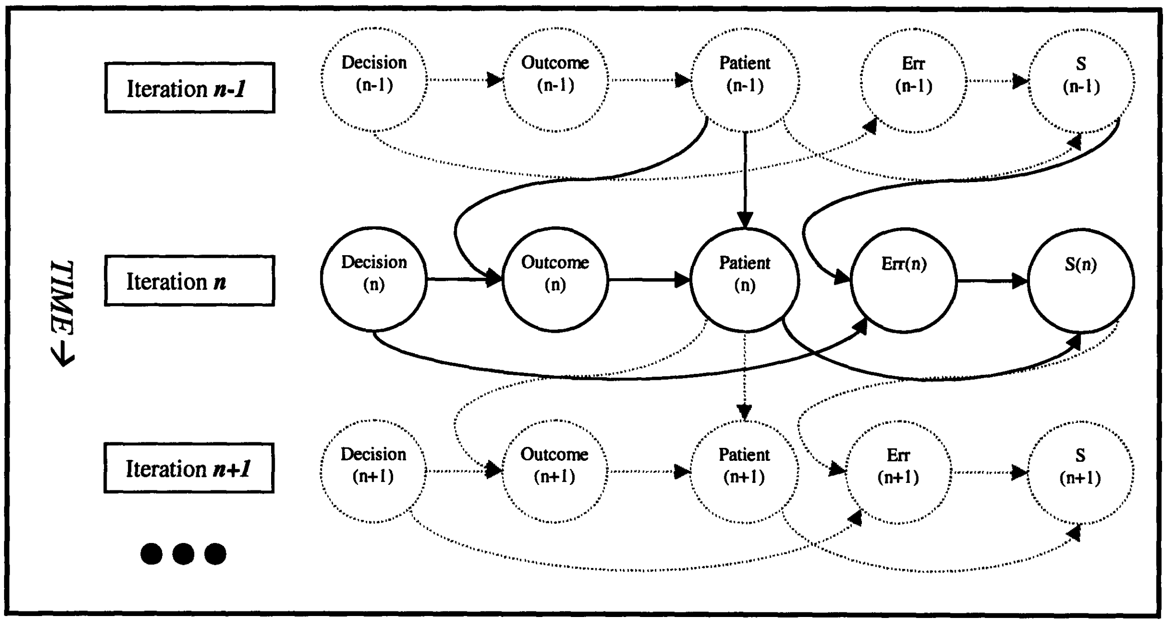


Figure 6- Influence Diagram of Key Model Elements at Iteration *n* (Active Simulation)

A decision is submitted by the user, and then evaluated to determine the outcome of the patient, as well as the quality of the user's decision. DSet(n) and the HALT subroutine are not depicted here; their presence is implied.

Subroutines of the Model

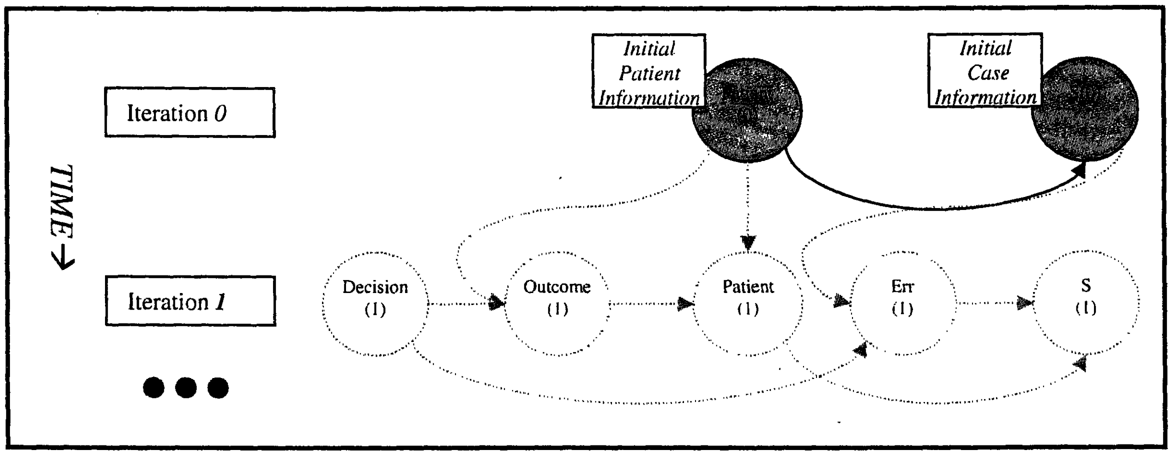
Here, the set of subroutines for the CDT model is defined. Collectively, these Subroutines will handle the task of updating the model. Accompanying each description is a reproduction of **Figure 6** which reflects the impact on the model elements after that step. A solid circle reflects an element whose attributes have been fixed, and a shaded circle reflects an element whose value may be fixed, but not necessarily.

Initialization:

In order to initialize the simulation, it is necessary to first initialize the status of the Patient, i.e. Patient(*0*). In traditional case studies, there is only one type of Patient considered, so this task is fairly straightforward. In a case where treatment strategy may vary greatly (perhaps due to the patient's age or time of presentation), a case author may want to simulate more than one type of Patient. The randomized choice of Patient(*0*) is conducted here.

At the beginning of a simulation, it is assumed that the user has not committed any tactical error. For various reasons, it may be necessary to raise or lower the error threshold for the user. If so, these adjustments are made in the initialization phase.

With Patient(*0*) set, the appropriate introductory patient and case information can be presented to the user and the value of S(*0*) is set. A copy of Patient(*0*) is created (for backtracking purposes) and the user can begin.



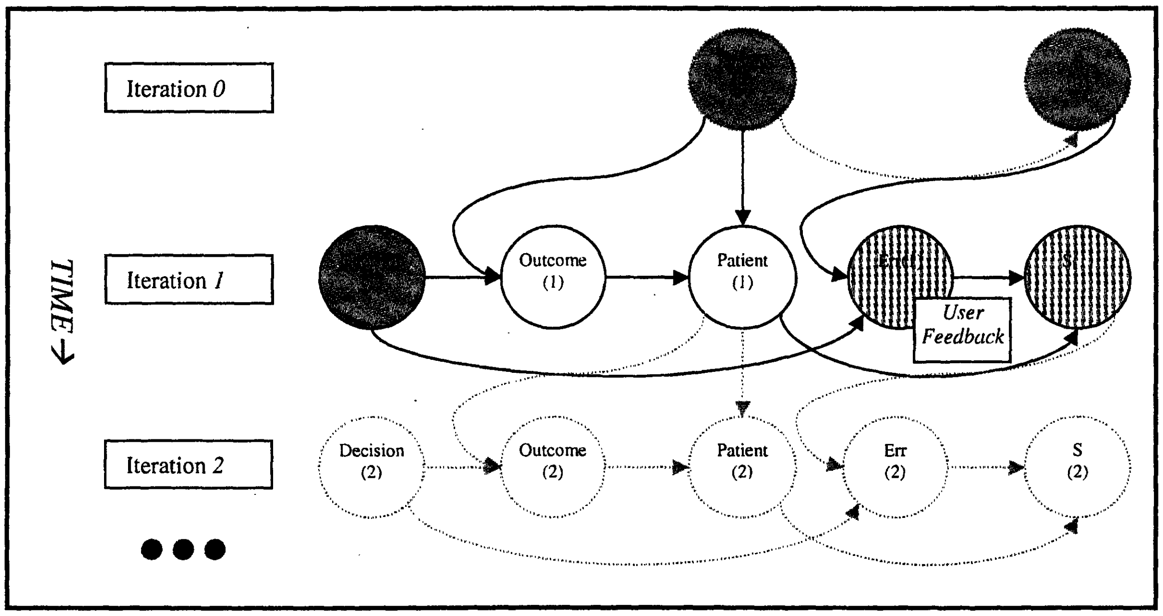
Preliminary Analysis (of Decision(*n*)):

This subroutine, along with the one described next, forms the first two parts of the Contextual Analysis Test (CAT) developed in **Chapter 2**. Recall that the conditioning events for this process depend on the decision made; in a worst-case scenario, there would be a separate subroutine of this type for each decision that the user can make. For all subroutines of this type, the purpose is the same, so the functionality need only be described once.

Given a decision, the first step is to determine whether or not the decision can be processed at all; from the pruning definitions, it is possible that the effects of some decisions may be UNDEFINED with respect to the active case. If this happens, Patient (***n***) becomes UNDEFINED and S(***n***) enters the HALT mode. Assuming that the decision can be processed, the key task is to determine the appropriate feedback to give to the user; this is done by analyzing Patient (***n-1***) for a matching conditioning event.

While determining the appropriate feedback, it is also appropriate to assess the severity of any perceived errors in judgment. If technical errors are detected, the value of Err(***n***) is increased. If Err(***n***) meets or exceed the system's error threshold, S(***n***) will be immediately put in HALT mode, otherwise, the model begins the process of *Outcome Simulation.*

Although determination of user error is not based on patient outcome, the status of Err(***n***) can not yet be fully defined, or "set". This is because Err(***n***) is also dependent on the previous state of the system, a check that occurs at the end of an iteration.



Outcome Simulation: (assuming the simulation is still ACTIVE)

This process incorporates the third part of the CAT, as the model must determine its ability to define a relevant outcome. Recall that S(***n-1***) denotes the most current status of the overall program; that is, at the time Decision(***n***) is made. During this subroutine, the parameters of S(***n-1***) are again analyzed for a particular type of conditioning event. This time, the conditioning event found is used to determine P(***n***| S(***n-1***)), the relative likelihood of outcomes for Decision(***n***), given S(***n-1***). The set of potential outcomes includes all relevant patient outcomes as well as any outcomes that trigger a pruning condition. If a pruning condition can occur with positive probability, the likelihood is captured by P(***n***| S(***n-1***)).

For individual outcomes, define the following:

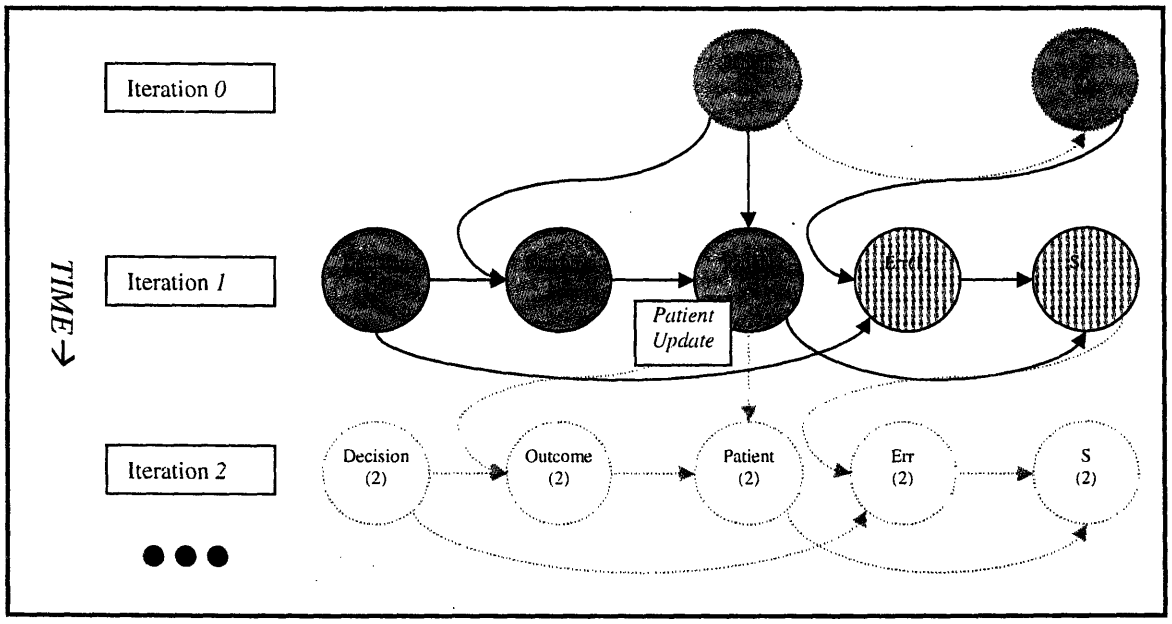
$$p_{i|s} = \Pr\{\text{Outcome } \boldsymbol{i} \text{ will occur}|\ S(\boldsymbol{n-1}) = S\}$$

Assuming ***R*** total outcomes, P(***n*** |S(***n-1***)) is then represented as a stochastic vector, that is, a vector of the form:

$$\{[\, p_{1|s}\, p_{2|s}\ \ \dots\ \ p_{R|s}]: \Sigma\, p_{i|s} = 1,\, p_{i|s} \geq 0\},$$

The values of $p_{1|s}$, $p_{2|s}$, etc. are determined by an appropriate set of conditioning events regarding the state of the patient and possibly other system factors, such as elapsed time. Unless the set of conditioning events is collectively exhaustive, P(***n*** |S(***n-1***)) will have to have a default value. For example, if no conditioning events can be found, P(***n***| S(***n-1***)) = [1, 0, ..., 0], with "Outcome 1" representing an UNDEFINED (pruning) outcome.

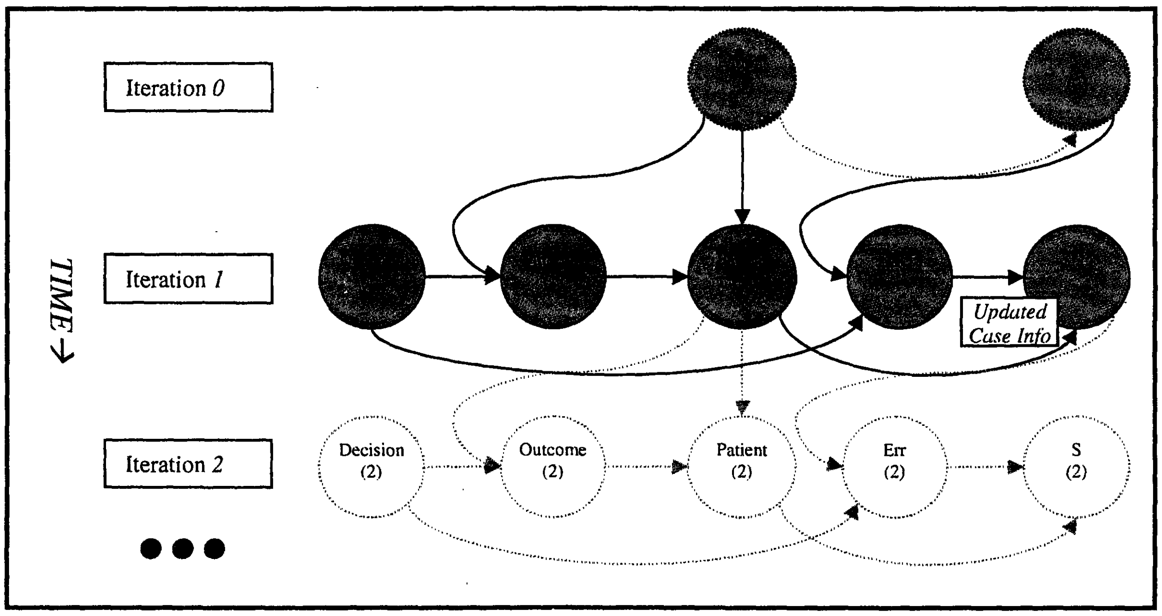
Once P(***n***| S(***n-1***)) has been determined, the corresponding CDF is used to map the possible outcomes to the numbers in [0,1]. A (pseudo-) random number on [0, 1] is used to select the simulated result, Outcome(***n***). If Outcome(***n***) represents a pruning condition, the system is set to HALT. Otherwise, the status of Patient(***n***) is updated; if any thresholds are violated as a result of the update, the system is put in HALT mode.



Systems Check: (assuming the simulation is still ACTIVE)

This process is essentially the fourth part of the CAT; in this subroutine, the overall state of the simulation is analyzed and any additional updates to the case are provided. For example, when there is a limit on the number of iterations, that condition is checked during this process. Also, there is some information that can only be provided to the user after a certain part of the case, or "milestone", has been reached; these milestones are tracked in this part of the program. This information is checked after every user input, regardless of the decision, which should be expected, as the fourth part of the CAT was stated to be global.

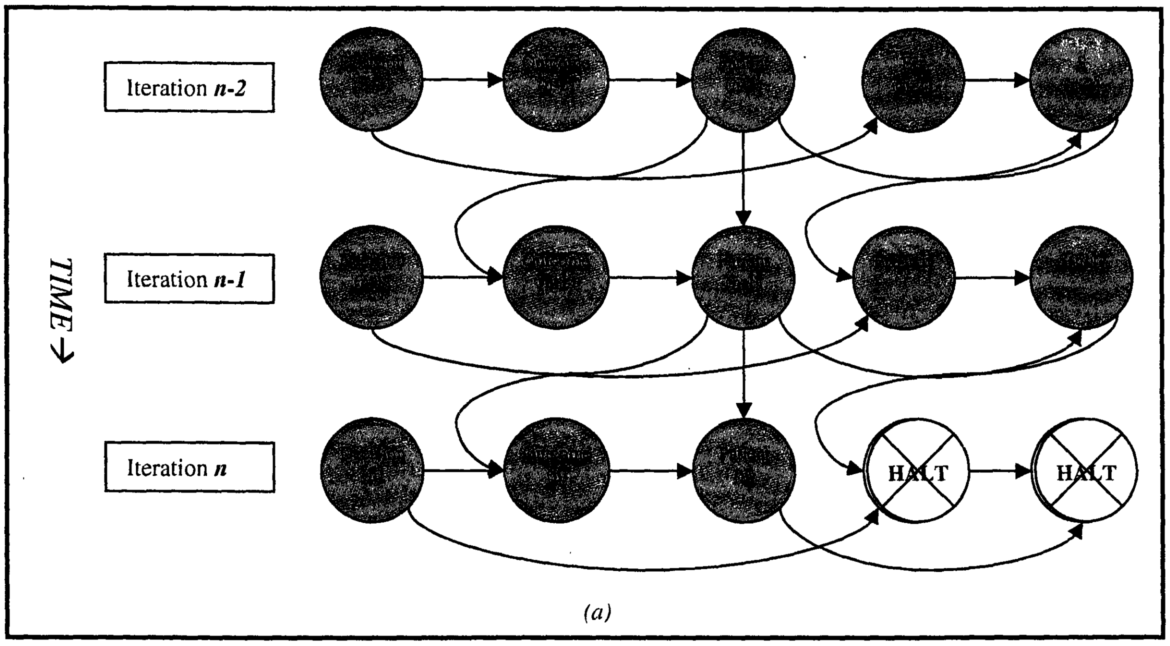
After these checks have been made, the value of Err(***n***) can be finalized. Since the status of Patient(***n***) is also understood, the state of the entire system at ***n***, S(***n***), is certain as well. If S(***n***) is ACTIVE, the model stores a copy of Patient(***n***) and Err(***n***), increments ***n***, and continues the simulation. If S(***n***) is in HALT mode, no copies are made and the model immediately enters the HALT subroutine.

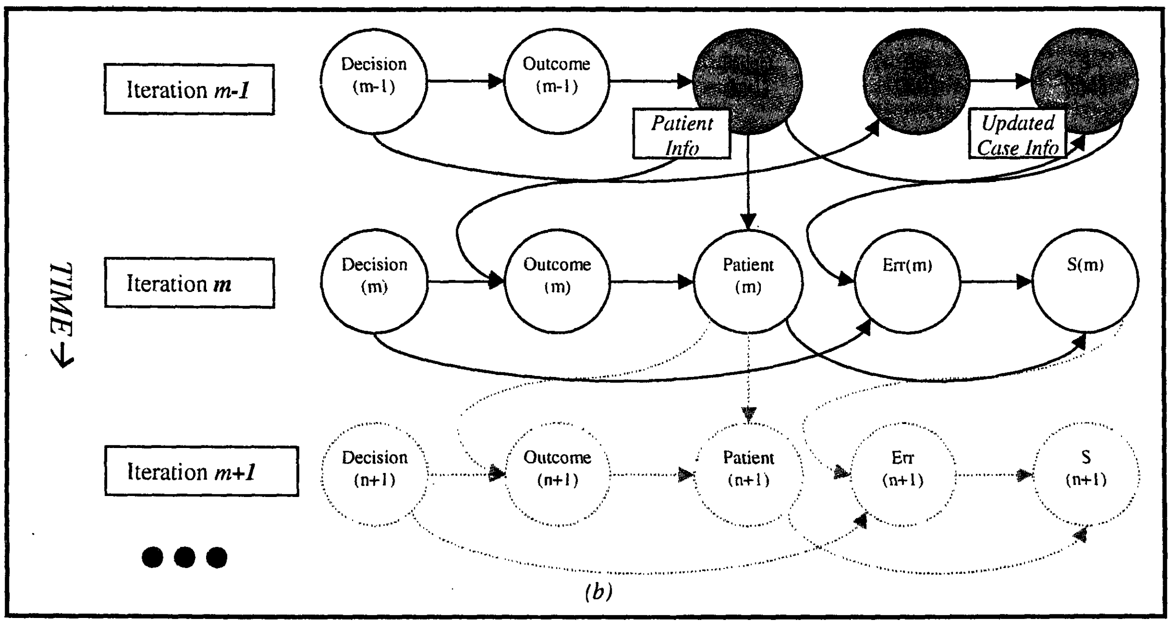


Halt Subroutine:

Once a simulation has been halted, the user is presented with all of the accumulated feedback regarding the case. This feedback usually will come from the case author; the feedback can also contain links to other reference materials. Once the feedback has been provided, the user can either exit the system altogether, or restart the simulation from any earlier point in time.

Figure (a) depicts a simulation that has been halted due to user error. Suppose the user would like to change a previous decision, Decision(***m***). Recall that the CDT model stores a copy of every previous system state; in order to allow this change, the copies of Patient(***m-1***) and Err(***m-1***) are accessed. The information therein is sufficient to reset the simulation, as shown in **Figure(b)**, and the user can now submit a new choice for Decision(***m***).





Process Algorithm:

This section concludes with an algorithm that summarizes the process flow of the model. The subroutines are denoted by **bold type**, and the text and media updates, which are provided to the user, are underlined.

```
{Initialization}

1. Determine the patient's initial state; call this Patient($0$)
   Initialize Err($0$);
   Let $n = 1$;

2. Given DSet($n$); user submits Decision($n$)

{Preliminary Analysis}

3. Using Decision($n$) and Patient($n-1$), collect any Feedback and/or determine the relative
likelihood of decision outcomes, P($n$| S($n-1$))

{Outcome Simulation}
4. Using P($n$| S($n-1$)), determine Outcome($n$), which updates Patient($n$)

5. Using Patient($n$), output Patient Information and update Err($n$)

(Systems Check}

6. Use System Parameters to output any new Case Information and determine S($n$),

7. Using S($n$), determine DSet($n+1$)

8. If S($n$) denotes an ACTIVE simulation,
        Then save a copy of S($n$)
                $n = n+1$; go to step 2
        Else, go to step 9

{Halt Subroutine}

9. Output all Feedback to the user.

10. The user will either quit the program or want to restart from decision $m \leq n$.
        If the user quits, then EXIT the program and close the application.

11. Set $n = m$ (Note: for all $m$, S($m$) is ACTIVE; otherwise, the system would have halted earlier.)

12. If  $m = 0$, go to step 1.

13. Retrieve Patient($m - 1$) and Err($m-1$); go to step 2.

14. Set $n = m$; go to step 2.
```

CHAPTER 4: IMPLEMENTATION - TECHNICAL NOTES

This section chronicles the steps taken to create the first implementation of the CDT model. The starting point was a traditional case study; after a review of the structure and content of the case, the educational objectives of the author are presented here to motivate the design of a stochastic analog. Next, the abstract elements of model (as defined in **Chapter 3**) are revisited in order to establish their relationships to the data structures present in the implementation. This section concludes with a series of visuals of the finished prototype.

About the case[4]

The case concerns a middle-aged man suffering from what is later revealed to be a listeria infection. Within the confines of the case, the user is required to recommend test procedures and antibiotic treatments on the patient's behalf; also, the user is presented with supplemental questions to test the user's general understanding and knowledge of the disease.

Dr. Sigal Yawetz, of the Brigham and Women's Hospital, provided the initial (deterministic) version of the case. The initial case study, which can be found in the Appendix, consisted of five questions; for context, they are paraphrased below:

- Given the patient state at arrival, what should the clinician (user) do <u>first</u>?
- Which antibiotics should be included in the initial treatment regimen?
- After viewing the result of a lumbar puncture, what is the likely cause?
- For patients allergic to ampicillin, which antibiotic is an appropriate substitute?
- Given the patient's lifestyle, how did the infection most likely occur?

Of the five, Questions 1 and 2 were predicted to generate the most incorrect answers. For Question 1, the author felt that early clinicians would likely delay treatment by performing some type of diagnostic; although this approach is taught regularly in medical school, the severity of this particular patient's condition requires immediate attention. Regarding Question 2, the author noted that experienced clinicians might neglect to include ampicillin in the initial treatment regimen. Ampicillin is specifically useful for treating listeria infections; the other antibiotics are not effective. Without knowing the cause of infection, it is not typical to assume the presence of listeria; however, best practice maintains that ampicillin should be provided just in case.

Because Questions 1 and 2 were considered the most critical elements of the case, the decision was made to introduce variability into the case at these points. In the original case, Question 1 only requires the user to declare the first step in patient care; the order of other actions is assumed. In contrast, the stochastic case study would require the user to provide the explicit order of patient care. For Question 2 of the original case, the user must submit the choice of antibiotics, but the patient's response is independent of the user's decision. In the stochastic case study, the behavior of the patient throughout the case is directly related to the user's choice of treatment. The core mistake in treating this particular patient is to delay ampicillin; the longer the patient goes without treatment, the more likely the patient is to become comatose, which would halt the simulation.

[4] The reader should note that the case used to develop this prototype is rather short and would not ordinarily require the tools developed in this paper. The model proposed in this paper would enable future case authors to construct much richer case environments; the implementation provided here serves primarily as a tangible, operational example of a case study created with the CDT model.

Representation of Model Elements

This section recalls the key elements of model design and relates them to the design of the prototype.

Decision(*n*), DSet(*n*) :

In the original case, there were five multiple-choice Questions presented to the user; accordingly, a user's potential decision was dependent on the Question being asked. In the stochastic case study analog, the set of possible user choices is also organized into five decision sets. The user's decision at iteration ***n*** (Decision(***n***)) comes from the union of two entities: the set of potential decisions at iteration ***n*** (DSet(***n***)); and the decision to HALT the simulation, which is always available. **Table 2** provides an overview of the decision sets for the prototype.

Questions 3-5 were not modified, but it is noted that Question 3 assumes that a lumbar puncture has been provided, and that Question 4 assumes that Ampicillin has been administered. Also, the decision was made to avoid "unlucky" patient scenarios; if the user navigates the case according to the author's recommendations, the simulation proceeds as originally written, without exception. In general, this is a very generous assumption to make, but, in this case, the recommended course of action is generally reliable and agreed upon. If a mistake is made regarding patient care (that is, regarding Questions 1 and 2), the simulation may or may not continue, depending on the severity of the error. Conceptual errors (such as misreading a lab result) do not impact the patient's health directly and are counted separately; the simulation would also halt if there were too many conceptual errors.

Name	Objective	Potential Decisions	Comments
***decideOrder*:**	Decide how next to treat the patient	Antibiotics	The preferred order is to administer antibiotics, provide plasma for the LP, and then conduct a CT scan. It is never advisable to conduct the LP without plasma
		CT- scan of the head	
		Lumbar puncture (LP) with fresh frozen plasma	
		Lumbar puncture without plasma	
decideTherapy:	Choose which antibiotics to administer to the patient	*(Any combination of the following)*	The recommend initial therapy is to provide ampicillin, ceftriaxone, and vancomycin. All three are recommended as initial therapy, but in the end, ampicillin is the only drug that will help this patient.
		Ampicillin	
		Ceftazidime	
		Ceftriaxone	
		Doxycline	
		Vancomycin	
analyzeGS:	Observe the gram stain from the lumbar puncture and determine the causative organism	Streptococcus pneumonia	The organism depicted is Listeria. All other choices are incorrect.
		Escherichia coli	
		Listeria monocytogenes	
		Neisseria meningitis	
reviseTherapy:	After an allergy to ampicillin is revealed, determine a good replacement antibiotic	Ceftazidime	Of the choices, trimetoprim is the only acceptble answer
		Erythromycin	
		Trimetoprim-sulfamethoxazole (bactrim)	
		Doxycycline	
determineCause:	After learning about the patient's lifestyle, determine the most likely cause of infection.	Prior travel to Vietnam and China.	The dietary habits are the only reasonable cause of listeria.
		A tick bite while in Cape Cod.	
		Dietary habits	
		A scratch sustained from a cat.	
		Rose thorn prick while gardening.	
HALT	The user wishes to stop the simulation	HALT	Always available; begins HALT subroutine

Table 2 - Decision Sets (DSets) defined for Prototype Case

Err(*n*):

Recall that Err(*n*) is a record of the cumulative tactical errors committed after ***n*** iterations. Once a decision is made, the first part of the contextual analysis test is performed to determine the validity of the decision. The validity of the user's decision is determined at the discretion of the case author.

If an error in judgment is detected, a penalty point is added to Err(***n***). Once a user receives two penalty points, the simulation is halted. The choice of two as a threshold is arbitrary and can be lowered (raised) in order to decrease (increase) the number of allowable scenarios. The penalty points that define Err(***n***) are the internal output of this contextual analysis.

The external output of error-checking is the author's commentary on the user's decision; i.e., the feedback the author wishes to provide the user. In implementation, this information is stored in an array called *Feedback*. As stated earlier, the information within *Feedback* is not revealed to the user until the simulation has been halted.

Patient(*n*), Outcome(*n*):

This implementation employs a combination of external and internal information to characterize the patient and the various outcomes that affect the patient. The external information consists of the narrative text and supporting visuals that guide the user through the simulation. The internal information consists of the data points and outcome probabilities that drive the behavior of the patient.

The external information is maintained within three separate arrays:

-*DecisionHistory*:	A text record of each decision made by the user.
-*CaseHistory*:	The full case narrative; a record of each user decision, plus every decision outcome and any supplemental information
-*MediaHistory*:	The set of supporting visuals; any slides, x-rays, etc. that have been displayed during the case.

During an active simulation, the *DecisionHistory* and *CaseHistory* arrays are updated once per iteration; the *MediaHistory* vector is updated as needed. As a convention, any initial information about the case is stored in the 0^{th} position (i.e., the head) of the array; consequently, at iteration ***n***, the length of each array is bounded above by $n+1$.

The array-based implementation allows the program to store the information regarding each iteration stage separately, but in chronological order. The contents of an array are accessible via a simple loop, providing a convenient method for displaying and updating the application interface. The contents of these arrays figure prominently in the finished prototype, which is shown in the following section.

For the internal information, there are two structures defined that are sufficient to capture the conditioning events that determine the behavior of patient. The first of these is an array (*NodeHistory*) that records the sequence of user decisions and patient outcomes. Each entry of this array is an ordered triple of indices representing a decision set, a decision, and the outcome. For example, the array information *NodeHistory*[3] = {1, 5, 2} has the following interpretation:

At iteration 3:
The user was presented with "Decision set 1";
The user chose "Decision 5", and
The result was "Outcome 2".

The other structure used to record internal information is a list of the medications that the patient is using at time ***n***. The antibiotics referenced in the "*decideTherapy*" decision set are considered ACTIVE or INACTIVE and are represented by a Boolean array called *RDActive*. The "RD" stands for "recurrent decision", which reflects the fact that the provision of antibiotics would have an ongoing impact on the patient's health. In the model that was constructed, the patient's underlying condition (infection by listeria) is assumed, so the structures mentioned are sufficient to simulate the patient's state; again, any available information is revealed to the user in text and images.

The Prototype Program

This section introduces the prototype program that was created from the CDT model. The program was written in the Java™ programming language; the NetBeans™ IDE was the primary tool used to develop, test, and debug the code. Notes regarding the code for this simulation can be found in the Appendix of this document; a CD containing this code is available as well.

The application interface is divided into five regions, which are continuously refreshed as the simulation progresses; see **Figure 7.** The top center panel (in white) is the primary information source for the user. The left and right panels serve as secondary information sources for the user. Directly below the white panel is the input panel that the user employs to maneuver the case. Also, a large button is provided at the bottom of the input panel, which allows us to halt the simulation at any time. The next several pages contains screen captures of the prototype at various states of execution; these screen captures serve to demonstrate the manner in which the external information described in the previous section is organized and presented to the user.

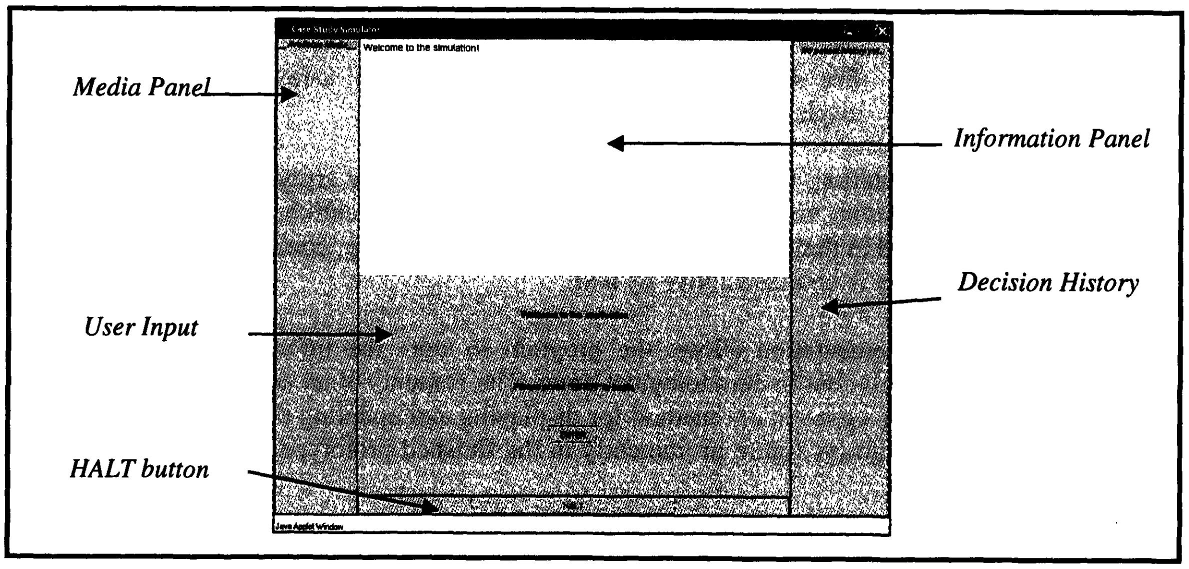


Figure 7 - Design of the Prototype Interface

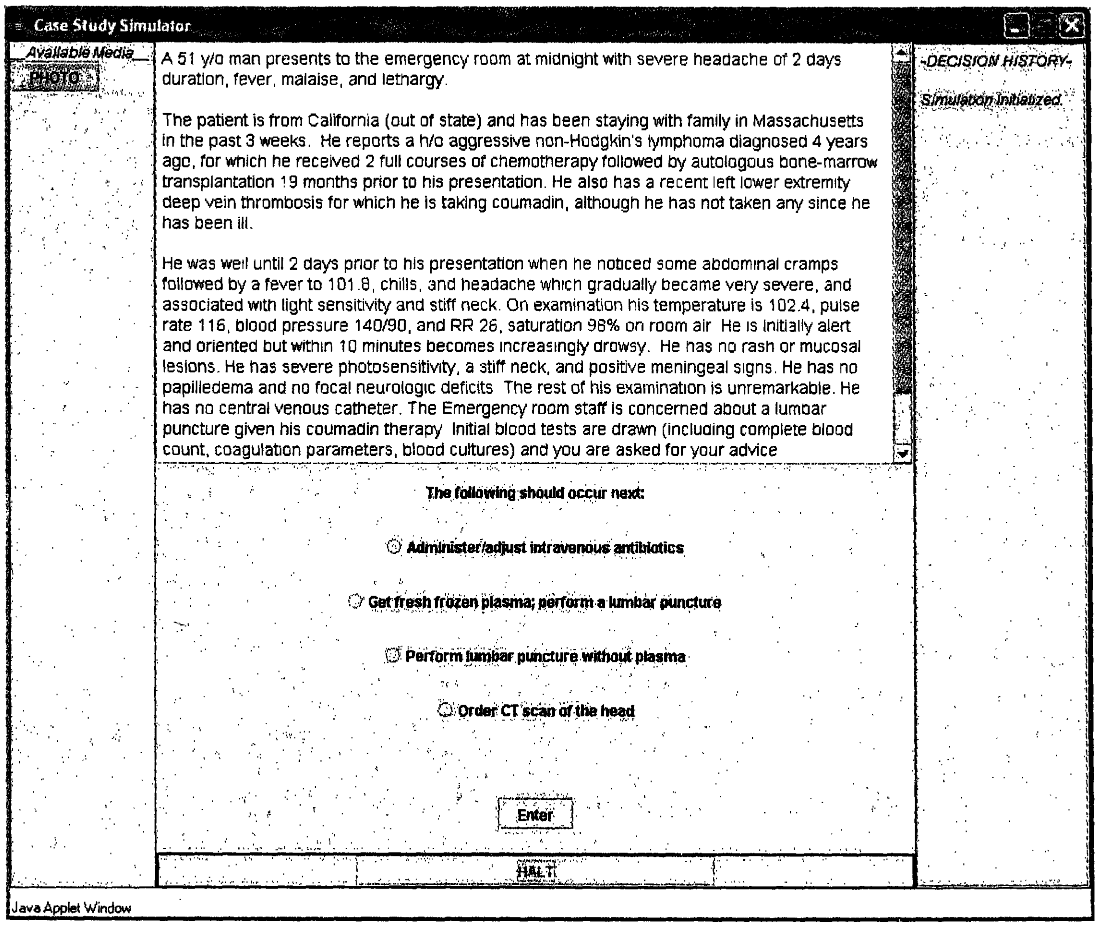
Active Case – Introduction

The first image depicts the look of the application shortly after the user decides to begin a simulation. Once a simulation has begun, the introductory case information appears in the central panel. During an active simulation, the case narrative is displayed here; as explained in the previous section, this narrative is stored in the *caseHistory* array.

During program execution, the panel at left is referred to as the "media panel"; its purpose is to provide links to any supplemental visual information for the case, including such as X-Rays, culture slides, photographs, etc. The media for this panel is stored within the *mediaHistory* vector.

The panel at the right provides a running history of the decisions made by the user during the simulation. This information is read from the *decisionHistory* array, described in the previous section. At the beginning of the simulation, this panel is empty; this is because no decision has been made by the user at this point.

The input panel has been updated to present the user with the first decision set, *decideOrder*. In an active simulation, the input panel will always presents the current decision set; i.e., the set of allowable decisions at the given state. The decision to HALT the simulation is always available to the user during an active simulation.



A 51 y/o man presents to the emergency room at midnight with severe headache of 2 days duration, fever, malaise, and lethargy.

The patient is from California (out of state) and has been staying with family in Massachusetts in the past 3 weeks. He reports a h/o aggressive non-Hodgkin's lymphoma diagnosed 4 years ago, for which he received 2 full courses of chemotherapy followed by autologous bone-marrow transplantation 19 months prior to his presentation. He also has a recent left lower extremity deep vein thrombosis for which he is taking coumadin, although he has not taken any since he has been ill.

He was well until 2 days prior to his presentation when he noticed some abdominal cramps followed by a fever to 101.8, chills, and headache which gradually became very severe, and associated with light sensitivity and stiff neck. On examination his temperature is 102.4, pulse rate 116, blood pressure 140/90, and RR 26, saturation 98% on room air He is initially alert and oriented but within 10 minutes becomes increasingly drowsy. He has no rash or mucosal lesions. He has severe photosensitivity, a stiff neck, and positive meningeal signs. He has no papilledema and no focal neurologic deficits The rest of his examination is unremarkable. He has no central venous catheter. The Emergency room staff is concerned about a lumbar puncture given his coumadin therapy Initial blood tests are drawn (including complete blood count, coagulation parameters, blood cultures) and you are asked for your advice

The following should occur next:

- ○ Administer/adjust intravenous antibiotics
- ○ Get fresh frozen plasma; perform a lumbar puncture
- ○ Perform lumbar puncture without plasma
- ○ Order CT scan of the head

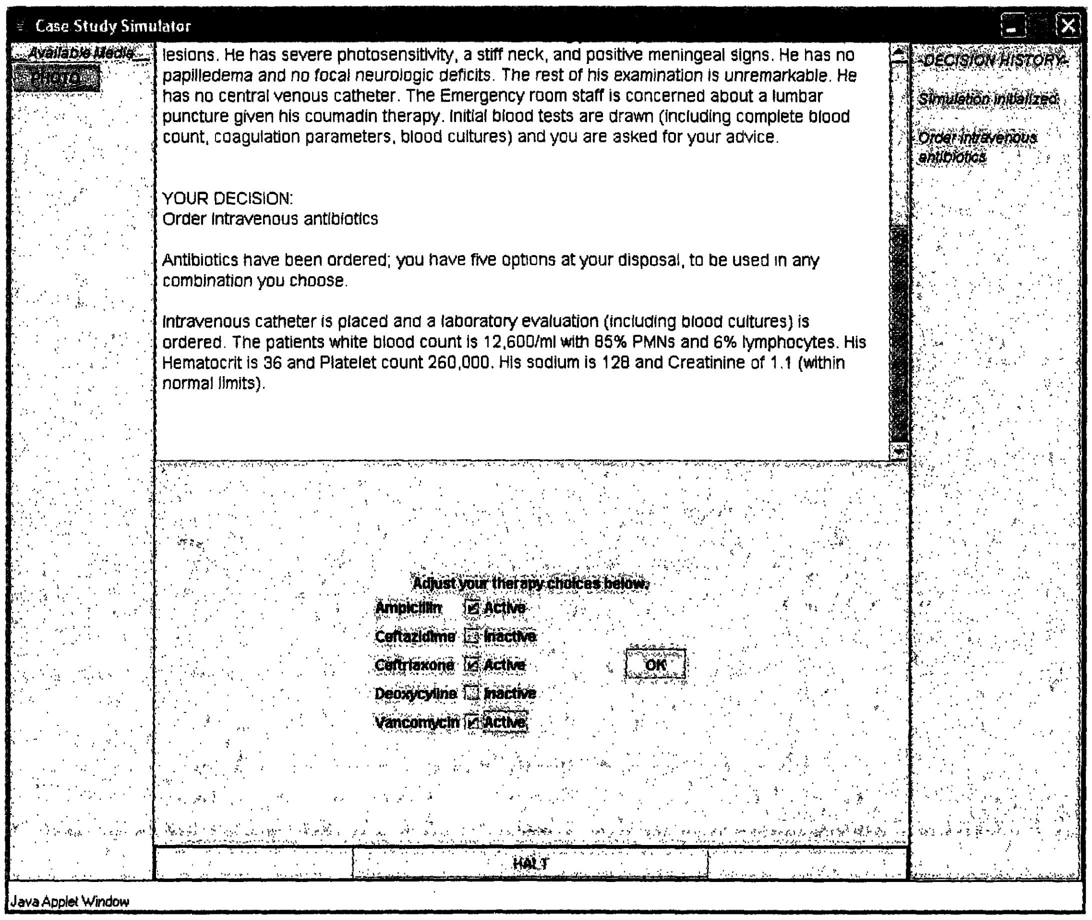
Enter

HALT

Active case – decideTherapy

The *decideTherapy* decision set presents the user with a list of the different treatments available. The user is allowed to start or stop use of the treatments throughout the duration of a simulaiton. Once a combination of therapies has been selected, the user must press the "OK" button in order for the simulation to store the information. From that point, the program will assume that the selected therapies remain active in the Patient for the remainder of the simulation, or until the user decides to alter the current (stored) therapy.

When the user submits the information, the results of the decision are shown and the user is given the next set of possible decisions. The simulation continues to operate in this way until the bounds of the simulation, as defined by the pruning criteria, have been reached. Any event defined as a pruning event serves as a signal to interrupt the user; if the patient dies, or if the case has been handled in an unsatisfactory manner, or if the patient's condition is indeterminable or otherwise outside of the educational scope of the simulation, the simulation halts. Of course, by pressing the red button, the user can opt to halt the simulation at any time.



lesions. He has severe photosensitivity, a stiff neck, and positive meningeal signs. He has no papilledema and no focal neurologic deficits. The rest of his examination is unremarkable. He has no central venous catheter. The Emergency room staff is concerned about a lumbar puncture given his coumadin therapy. Initial blood tests are drawn (including complete blood count, coagulation parameters, blood cultures) and you are asked for your advice.

YOUR DECISION:
Order intravenous antibiotics

Antibiotics have been ordered; you have five options at your disposal, to be used in any combination you choose.

Intravenous catheter is placed and a laboratory evaluation (including blood cultures) is ordered. The patients white blood count is 12,600/ml with 85% PMNs and 6% lymphocytes. His Hematocrit is 36 and Platelet count 260,000. His sodium is 128 and Creatinine of 1.1 (within normal limits).

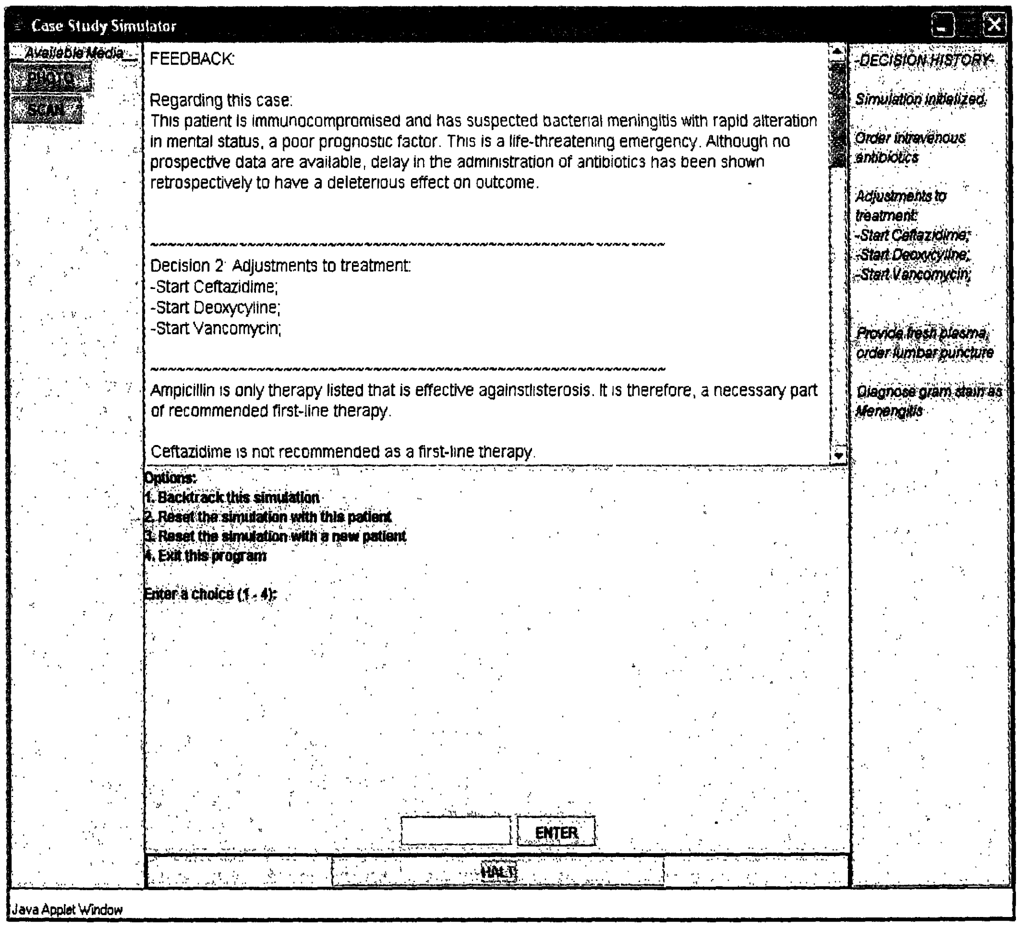
Adjust your therapy choices below:

Ampicillin	✓	Active
Ceftazidime	☐	Inactive
Ceftriaxone	✓	Active
Deoxycyline	☐	Inactive
Vancomycin	✓	Active

Halt simulation screen:

When the simulation has been halted, the central panel and the input panel are modified as needed to guide the user forward. Immediately after a simulation has been halted, the main information panel is used to present feedback to the user; this includes commentary regarding the case, as well as any other guidance for the user. As mentioned before, this information is stored in the *Feedback* array.

The input screen is updated to provide the user with four options. Option 4 exits the entire program. Options 2 and 3 both reset the simulation from the beginning, with a slight difference. If the patient's initial condition is always the same at the start of a simulation, options 2 and 3 will be the same. However, if we allow for a randomized initial patient, option 2 will allow the user to restart the simulation with that same particular patient. Option 1 represents the backtracking option, which would allow the user to review the previous decision sequence and try a different strategy.



FEEDBACK:

Regarding this case:
This patient is immunocompromised and has suspected bacterial meningitis with rapid alteration in mental status, a poor prognostic factor. This is a life-threatening emergency. Although no prospective data are available, delay in the administration of antibiotics has been shown retrospectively to have a deleterious effect on outcome.

~~~~~~

Decision 2: Adjustments to treatment:
-Start Ceftazidime;
-Start Deoxycyline;
-Start Vancomycin;

~~~~~~

Ampicillin is only therapy listed that is effective againstlisterosis. It is therefore, a necessary part of recommended first-line therapy.

Ceftazidime is not recommended as a first-line therapy.

Options:
1. Backtrack this simulation
2. Reset the simulation with this patient
3. Reset the simulation with a new patient
4. Exit this program

Enter a choice (1 - 4):

-DECISION HISTORY-

Simulation initialized.

Order intravenous antibiotics

Adjustments to treatment:
-Start Ceftazidime;
-Start Deoxycyline;
-Start Vancomycin;

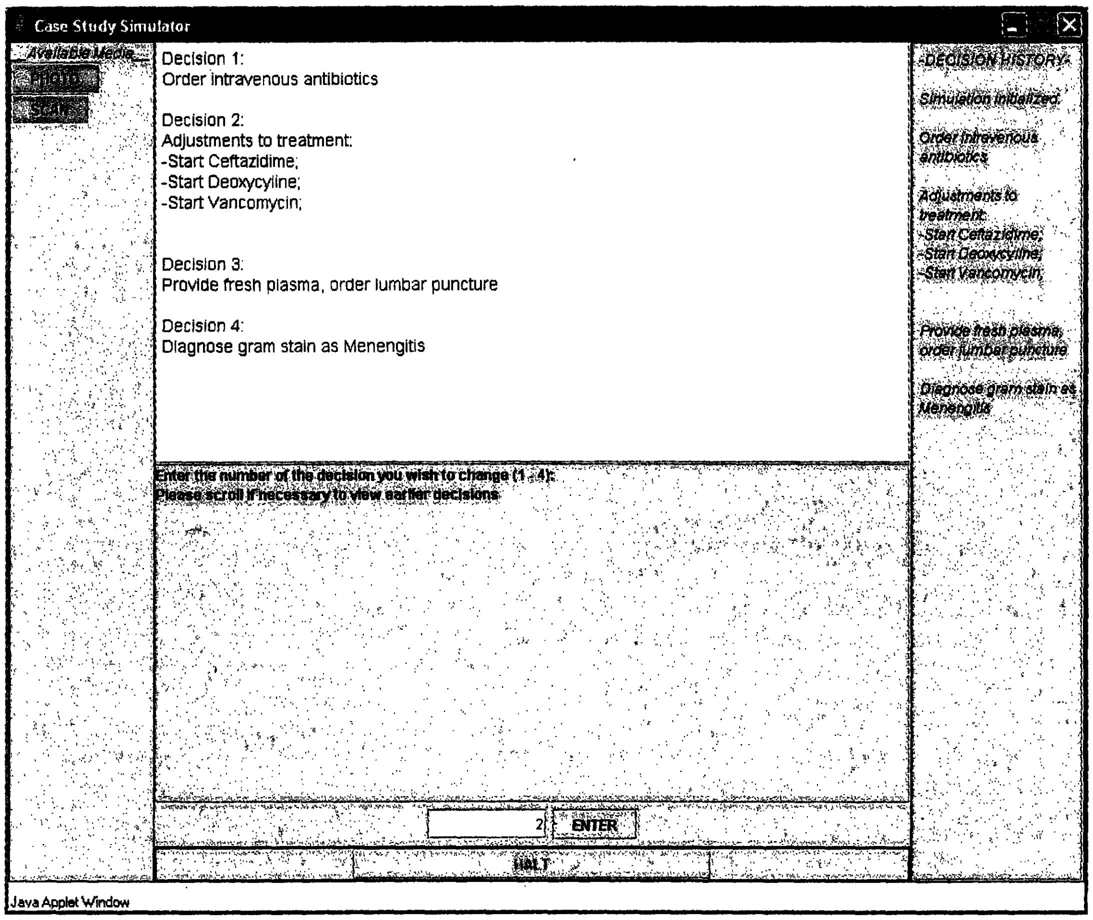
Provide fresh plasma, order lumbar puncture

Diagnose gram stain as Menengitis

Backtrack screen:

The final element of functionality discussed here is the backtracking option. Recall that this is the feature that allows the user to restart the halted simulation from an arbitrary point in the past; as a result, the user has an opportunity to immediately apply the lessons learned from feedback and roll back to an earlier decision.

As can been seen here, the information screen updates again; this time, to show the history of all decisions made in this case. This is precisely the same information that can found in the panel at right; i.e., the information from *decisionHistory.* At this point, the user has the opportunity to change any decision that was made and restart the case from that point. Events occurring after that point are erased, and all events before that point are preserved.



Decision 1:
Order intravenous antibiotics

Decision 2:
Adjustments to treatment:
-Start Ceftazidime;
-Start Deoxycyline;
-Start Vancomycin;

Decision 3:
Provide fresh plasma, order lumbar puncture

Decision 4:
Diagnose gram stain as Menengitis

Enter the number of the decision you wish to change (1 - 4):
Please scroll if necessary to view earlier decisions

2

-DECISION HISTORY-

Simulation initialized.

Order intravenous antibiotics

Adjustments to treatment:
-Start Ceftazidime;
-Start Deoxycyline;
-Start Vancomycin;

Provide fresh plasma, order lumbar puncture

Diagnose gram stain as Menengitis

CHAPTER 5: IMPLEMENTATION – USER IMPACT:

As stated in the introduction, the purpose of developing this model was to create a more engaging, durable approach for learning this material. The approach presented here lead to three major changes to the existing model. The fundamental change was to provide a model that supports stochastic (i.e., probabilistic) patient behavior. This modification prompted the second major change: the separation of user feedback and patient status. The separation of feedback is accomplished by delaying the presentation until the end, when all feedback is presented at once. With all of this information being delivered at once, it was necessary to enable users to identify and isolate specific errors in decision-making; the backtracking feature was developed to allow this functionality.

This ideas presented here represent a mathematics-based approach proposed by engineers; the inherent risk in promoting the CDT model would be that the proposed modifications would ultimately be of little use to the intended end user, the medical professional. With this thought in mind, it is instructive to get perspective from members of the medical community on stochastic case studies and the CDT approach and potential impact of developing future cases in this manner. Of particular interest are the perceived advantages and disadvantages of the modifications that help define the CDT model.

In order to get this perspective, a review of the prototype was solicited from the primary medical contact and original case author, Dr. Sigal Yawetz. Despite extensive familiarity with traditional case studies, Dr. Yawetz had not considered this style of case study prior to this project. Upon completion of the prototype, Dr. Yawetz was invited to use the system and asked for feedback regarding its functionality. Supplemental feedback was also provided by Ms. Melinda Cerny, who has served as project manager on a series of educational websites for medical professionals. As of this writing, she is preparing to launch a website on Transplant Infectious Diseases; the website project provided the early motivation for the work presented herein. Given Ms. Cerny's familiarity with the development and functionality requirements for these projects, her assessment of the model is of special interest for identifying any barriers to implementation.

On the use of uncertainty

The use of uncertainty in the CDT model represents the core departure from the traditional case study method; accordingly, it was predicted that this part of the approach would be the most controversial. Contrary to expectation, when the model was demonstrated for an audience, user comments regarding this feature of the model were extremely positive. The project manager surmised that from a pedagogical standpoint, the notion of providing cases with alternative outcomes would be a very welcome addition to her project websites. After being presented with the CDT model, she was "excited" about the future prospects for these cases; also, she expressed interest in incorporating the existing prototype into the current project.

Overall, the case author was also very supportive of the CDT approach. Throughout the development phase, she remarked that clinicians, as case authors, are not used to thinking about cases in a probabilistic manner; the narrative is generally drawn from a particular patient interaction, and the specifics of that interaction are treated as absolute truth with respect to the case. However, she opined that, over time, stochastic case studies would serve as a valuable proxy for clinical experience, and that future case authors would benefit from thinking through alternative scenarios as a case is developed.

Other CDT functionality

Other features introduced in the CDT model were also well received. The separation of program simulation and user feedback implies that the user is given minimal assistance while navigating a case. The case author noted that users of the system would be unaccustomed to the concept of delayed feedback, and would probably prefer the traditional method of instant feedback for decision-making. Still, because the CDT model is a better approximation of patient care, she felt it more than reasonable to sacrifice comfort for realism and develop more cases using the CDT model.

The ability to backtrack past decisions was developed as a means of minimizing the negative effects of delaying feedback, and the added functionality was noted in the implementation. In constructing the model, it was reasoned that the ability to backtrack would be sufficient to allow the user to address specific development needs, just as in the traditional model. After viewing the prototype, the project manager agreed with this view, even suggesting that, from a functional standpoint, the CDT model "sacrifices nothing" in providing a rich learning environment for clinician training.

Notes on Future Implementation

In addition to providing feedback on the functionality of the CDT model, colleagues were also asked to share their views regarding effective implementation and adoption of the CDT model. While noting the benefits of stochastic cases, it was made clear that the CDT model should not be viewed as a replacement for traditional case studies. Instead, the consensus opinion was to treat stochastic cases as part of a multi-pronged approach to clinician training.

With regard to clinician training websites, a stated goal is to develop the content necessary to create and support clinician certification. The case author noted that the variability of a stochastic case would ultimately hinder the ability to assess performance across a group of users. The presence of uncertainty allows for dissimilar experiences across a group, a concept contrary to the notion of standardized group assessment, which generally assumes a common test environment. With this in mind, the case author promoted the use of stochastic case studies as a supplemental certification tool. In other words, a user would be required to attempt and complete stochastic cases as a means of experimental learning, but users would never be penalized for substandard performance.

An understated, but obvious, disadvantage of the CDT model is the increased workload for the case author. As the complexity of a stochastic case increases, so does the amount of case commentary and event conditioning. It was reasoned that the extra work required to build stochastic cases would slow down the production of new cases as well as dissuade potential authors from embracing the CDT model. For this reason, project manager suggested that the number of stochastic cases be kept to a relative minimum, with perhaps no more than two or three on a website. In this format, the user could gain basic skills through traditional case studies, and use the stochastic cases as a more rigorous self- test of their decision-making ability.

CHAPTER 6: REVIEW AND FUTURE WORK

This paper concludes with an argument for continued exploration in this area of research and some suggestions on how to proceed. After reviewing the breadth of the work presented here, the long-term impact of the CDT approach is considered, for program users and case authors. Given this assessment, there are a number of potential avenues for continued research in this area.

Review of Presented Research

Given the notion of a case study, the challenge posed was to extend the functionality in a manner that would best serve the needs of an interactive medical website. The goal was to create an online "living document" that would accessible to clinicians around the world and extensible to future case modifications by practicing clinicians.

In the course of analysis, a structure was developed - the Compressed Decision Tree (CDT) - that facilitates simulation over a set of potential outcomes despite imperfect knowledge of all possible outcomes. This simulation is achieved using a four-step Contextual Analysis Test that uses information about the current state of the system to determine decision outcomes and future model behavior. The development of the CDT structure provided a basis for an alternative approach to structuring case studies, referred to here as the CDT model.

Assessment of Long-Term Benefits

The CDT model provides a method of producing stochastic case studies, which, as indicated by the response to the prototype, are a fairly novel approach to training clinicians. More important than the novelty of the approach is the common perception that stochastic cases could provide a significant contribution to the field of web-based medical education.

This optimism is bolstered by the various benefits associated with stochastic case studies. The end user is provided with a dynamic learning environment and tailored guidance, providing a more personalized experience. The use of uncertainty in a case provides the user with a variety of patient scenarios, which increases the likelihood that a case is accessed more than once. This notion of reusable educational content is of immediate relevance to creators of educational websites, many of whom depend on the use of engaging, enduring content

Perspectives on Future Research

In anticipation of a broader implementation, it should be made clear that there are multiple avenues for continued research and development of the CDT model. Perhaps the most pressing among these is the creation of a case development interface for case authors. The case presented in the prototype had to be manually converted from a traditional case, a process that required extensive editing and computer programming. In contrast, the functionality of the development interface would allow case authors to develop stochastic cases directly, without explicit use of a programming language.

Looking ahead, it would also be worthwhile to revisit the CDT model and discover areas for improvement. The purpose of developing the CDT model was to facilitate and promote the development of a more engaging and effective interactive experience; regardless of the means, future research should focus on making user-centric case studies and educational tools a reality.

(This page is left intentionally blank)

CHAPTER 7: REFERENCES

[1] "Instructional Strategies for Active Learning: Case-Based Learning/Case Studies". [Online Teaching Resource] The Getty Conservation Institute. Accessed January 2005. Available from http://extranet.getty.edu/gci/teaching/activelearning_case.html.

[2] Muldoon, M.F., Barger, S.D., et al., "What are "quality of life" measurements measuring?" *BMJ (British Medical Journal),* 316:542-545, 1998.

[3] Raiffa, H. *Decision Analysis: Introductory Lectures on Choices under Uncertainty*: 250-255. Reading, MA: Addison - Wesley; 2nd Printing, 1970.

[4] Sklar, B. "The Current Status of Online Continuing Medical Education." Master's Thesis, University of California, San Francisco, 2000.

[5] Szolovits, P. "Uncertainty and Decisions in Medical Informatics." *Methods of Information in Medicine*, 34: 111-121, 1995.

(This page is left intentionally blank)

APPENDIX A: TEXT DRAFT OF CASE MATTER[5]

The following text is a draft copy of the case study narrative written for the prototype case. For the case questions, the suggested responses are denoted with an asterisk. Comments for the user and narrative feedback are italicized.

"Patient with Listeria"

Author: Dr. Sigal Yawetz, Brigham and Women's Hospital, Boston MA

A 51 y/o man presents to the emergency room at midnight with severe headache of 2 days duration, fever, malaise, and lethargy.

The patient is from California (out of state) and has been staying with family in Massachusetts in the past 3 weeks. He reports a h/o aggressive non-Hodgkin's lymphoma diagnosed 4 years ago, for which he received 2 full courses of chemotherapy followed by autologous bone-marrow transplantation 19 months prior to his presentation. He also has a recent left lower extremity deep vein thrombosis for which he is taking coumadin, although he has not taken any since he has been ill.

He was well until 2 days prior to his presentation when he noticed some abdominal cramps followed by a fever to 101.8, chills, and headache which gradually became very severe, and associated with light sensitivity and stiff neck. On examination his temperature is 102.4, pulse rate 116, blood pressure 140/90, and RR 26, saturation 98% on room air. He is initially alert and oriented but within 10 minutes becomes increasingly drowsy. He has no rash or mucosal lesions. He has severe photosensitivity, a stiff neck, and positive meningeal signs. He has no papilledema and no focal neurological deficits. The rest of his examination is unremarkable. He has no central venous catheter. The Emergency room staff is concerned about a lumbar puncture given his coumadin therapy. Initial blood tests are drawn (including complete blood count, coagulation parameters, and blood cultures) and you are asked for your advice.

1. The following should be done next:

    1. Head CT to exclude the presence of an intra-cranial lesion
    2. Empiric intravenous antibiotics*
    3. Fresh frozen plasma, followed by lumbar puncture (CSF examination)
    4. Lumbar puncture (CSF examination) without fresh frozen plasma
    5. None of the above

[Answer narrative] This patient is immunocompromised and has suspected bacterial meningitis with rapid alteration in mental status, a poor prognostic

[5] This material is used with the permission of the case author and should not be redistributed or used in any part without the author's consent.

factor. This is a life-threatening emergency. Although no prospective data are available delay in the administration of antibiotics has been shown retrospectively to have a deleterious effect on outcome.

[For answers 3 or 4 LP]: Although the use of antibiotics may affect the accuracy of CSF gram stain and culture result it is crucial that ant microbial therapy not be delayed due to an inability to immediately perform a spinal tap. Given the concern about this patient's coumadin therapy a CSF examination could not be performed promptly. However appropriate antibiotics should be immediately provided.

[For answer 1, ct] A common mistake is to delay antimicrobial therapy and CSF examination by obtaining a head CT to exclude an intracranial lesion. However, it has been shown that this practice has led to significant delays in therapy. It has been shown that the likelihood of hernia ion is low in the absence of papilledema and focal neurological findings (<1.2% as compared with 12%) as in this case. In cases where the risk of herniation appears to be high, antimicrobial therapy should be initiated before obtaining a head CT, even if this will delay a lumbar puncture.

2. Empiric antibiotic choice in this case should include the following agents (mark all that apply):

- ☐ Doxycycline
- ☐ Ceftriaxone*
- ☐ Ceftazidime
- ☐ Vancomycin*
- ☐ Ampicillin*

[Answer narrative]

General principles: Factors influencing the choice of empiric antibiotics for suspected purulent meningitis include:

Ability to penetrate the blood-brain-barrier in the presence inflammation

Bactericidal activity, even in a purulent environment (with an acidic pH_)

Activity against microorganisms specific to the host's age, medical history immune status

Specific agents:

Doxycycline – is a bacteriostatic agent with poor penetration of the blood-brain barrier. It is not recommended as first line empiric therapy for meningitis.

Ceftriaxone - a third generation cephalosporin, is the beta-lactam of choice in the empiric treatment of meningitis in adults. It penetrates the BBB well and has activity against the major pathogens of bacterial meningitis including most strains of Streptococcus pneumoniae including penicillin-resistant strains, Neisseria meningitis, most Escherichia coli isolates, and Haemophilus influenza with the notable exception of Listeria.

Ceftazidime, a third generation cephalosporin, is less active against penicillin-resistant pneumococci than cefotaxime and ceftriaxone. Its use is reserved for patients with neutropenia, cerebrospinal fluid shunts, and neurosurgical procedures or trauma due to the higher risk of resistant gram negative organisms in these populations.

Vancomycin - is inconsistent in its penetration of blood-brain-barrier. However given the increasing rates of beta-lactam resistance in Streptococcus pneumoniae, it is used with third generation cephalosporins in area where high-level penicillin resistance is seen. Vancomycin is continued if resistant Streptococcal pneumoniae is suspected or isolated, and the MIC to third generation cephalosporins is >0.5 μg/mL

Ampicillin – is added for empiric coverage for Listeria monocytogenes, a common cause of CNS infections in the transplant patient and in patients with lymphoprolifertive diseases. Third generation cephalosporins such as ceftriaxone have no activity against this organism. Listeriosis may occur in the early or late post-transplant period, and a common mistake is to omit Ampicillin from the initial empiric regimen in those patients. Listeria should be suspected and empirically treated in any meningitis patient older then 50 or immunocompromised.

Intravenous catheter is placed and laboratory evaluation (including blood cultures) is drawn. Vancomycin, ceftriaxone, and ampicillin are provided. The patients white blood count is 12,600/ml with 85% PMNs and 6% lymphocytes. His Hematocrit is 36 and Platelet count 260,000. His sodium is 128 and Creatinine of 1.1 (within normal limits). Fresh frozen plasma is ordered, but the INR is 1.1 and lumbar puncture is performed. Analysis of the cerebrospinal fluid (CSF) shows 1580 WBC with 96% polymorphonuclear cells. The CSF protein concentration was 98, and the CSF glucose concentration was 45 (serum 116). The gram stain reveals the following organisms [show picture of large gram positive rods].

3. The most likely causative organism is:

1. Streptococcus pneumonia
2. Escherichia coli
3. Listeria monocytogenes*
4. Neisseria meningitis

A CT scan of the head is unremarkable. In the CT scanner the patient is noted to be wheezing and has hives. He is more confused and requires intubations and mechanical ventilation. The patient wife and brother arrive and report a penicillin allergy, although he has received Ceftazidime in the past for fever and neutropenia without problem. He has no other known drug allergies.

4. Which of the following agents is an alternative to ampicillin for the suspected organism:

1. Ceftazidime
2. Erythromycin
3. Trimetoprim-sulfamethoxazole (bactrim) *
4. Doxycycline

[Answer Narrative]

Penicillin-allergic patients with listeriosis may be desensitized to penicillin if this is feasible. In this critically ill patient this may not be possible.

> *Ceftazidime - cephalosporins are inactive in vitro against listeria and have been shown to be clinically ineffective.*
>
> *Erythromycin has in vitro activity against listeria, although is bacteriostatic. It is not recommended as a second line agent for CNS listeriosis.*
>
> *Trimethoprim-sulfamethoxazole is penetrated the blood-brain-barrier well, has is bactericidal against Listeria, and has been shown to be clinically effective at high doses (20mg/d of trimethoprim divided into 4 daily doses).*
>
> *Doxycycline has in vitro activity against listeria, although is bacteriostatic. Although it is used for other CNS infections, docycycline is not recommended as a second line agent against systemic or CNS listeriosis.*

The following additional information is obtained from the patient's wife: the patient is a freelance journalist and travels throughout the country usually by air. He has not traveled outside of the US since his illness, but has previously been to Vietnam, China, Brazil, Mexico, Hawaii, Canada, France and Germany. He had no recent respiratory or gastrointestinal infections and did not use any antibiotics. He often forgets his coumadin when traveling. He does not smoke or drink, and has no history of injection drug use. No other family member is currently ill. He is a vegetarian. The family ate mostly at home. He has a dog and his hosts have two cats. The family spent the past three weekends on Cape Cod but had not experienced tick bites. He sustained a rose thorn prick while at his brother's garden.

The CSF cultures remain negative, but within 30 hours the blood cultures reveal a gram-positive rod, later identified as Listeria monocytogenes.

5. Which of the following may have contributed to his illness?

1. Prior travel to Vietnam and China.
2. A tick bite while in Cape Cod.
3. Dietary habits.*
4. A scratch he sustained from his hosts' cat.
5. Rose thorn prick while gardening.

(This page is left intentionally blank)

APPENDIX B: PROGRAMMING NOTES FOR PROTOTYPE

This section provides supplemental information regarding the code (.java files) for the prototype. The actual code is viewable on the included CD. As a guide, some basic notes about the classes are provided below. With the exception of the Viewer class, the classes listed below were developed by the author.

For those who wish to test a demo of the program, please try the online version, located at http://orc-pumba.mit.edu/ljc/applet/Simulation.html. (Note: The computer must be able to run Java(TM) applets.)

If the link above becomes inactive, please notify the author: ljchandler@alum.mit.edu

CLASSES OF THE PROTOTYPE

Decision

This class is used to define a set of general behaviors associated with a user-submitted decision. For the prototype, this class contains the functions necessary to generate pseudo- random outcomes.

GUI

The GUI class is the key executable; it is responsible for relaying information from the user to the underlying model and vice versa. Settings for the physical dimensions and appearance of the user interface are also located here.

HLink

The HLink class was developed as a compact means of storing information regarding past decisions and outcomes. A new HLink object is created for every decision the user makes. An HLink is structured as an ordered triple with elements representing the Decision Set presented to the user, the Decision chosen by the user, and the resulting Outcome.

MediaButton

The MediaButton class is invoked by the GUI class. When a new image is available for viewing, a MediaButton object appears on the user interface, as a small button. When the user presses the button, the requested image is displayed in a (Viewer) window.

Model

The Model class is the logical center of the case module. The Compressed Decision Tree and all Contextual Analysis tests reside in this class. Additionally, all commentary relating to the case is kept here, including user feedback.

Node

The Node class contains the functions necessary to create storable copies of the simulation in progress. Should a user decides to backtrack to an earlier point in the case, a Node structure will contains all of the information needed to reset the program to the desired state.

Viewer

The Viewer class is used to facilitate the viewing of images in the user interface. The code for the Viewer class was provided by a department colleague. The apparent author of the code is Marco Schmidt; further information about the class can be found online at the following address: http://www.geocities.com/marcoschmidt.geo/java-load-image-toolkit.html