Scientia: An End User Development Environment for Decision Support Systems

by

Clayton Tyler Sims

B.S., Massachusetts Institute of Technology (2008)

Submitted to the Department of Electrical Engineering and Computer Science

in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author ....
Department of Electrical Engineering and Computer Science
August 21, 2009

Certified by ....
Robert C. Miller
Associate Professor
Thesis Supervisor

Accepted by ....
Dr. Christopher J. Terman
Chairman, Department Committee on Graduate Theses

Scientia: An End User Development Environment for Decision Support Systems

by

Clayton Tyler Sims

Submitted to the Department of Electrical Engineering and Computer Science
on August 21, 2009, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

This thesis describes an end user programming environment that allows non-programmers to create decision support protocols for use on electronic devices. User centered design techniques were followed to identify the difficulties encountered by users when attempting to create complex protocols, specifically addressing the problems of the scale, complexity, and specificity required for a protocol to be effectively used. The result is a highly usable desktop client graphical user interface which can create protocols that can be exported in portable formats. A summative user study was conducted on the finished software in order to evalute its success in enabling non-programmers to author protocols.

Thesis Supervisor: Robert C. Miller
Title: Associate Professor

Acknowledgments

I'd like to thank all of the experts who took time out of their schedules to meet with me and discuss protocols or suggest protocols for me to analyse including but not limited to Dr. Marc Mitchell, Neal Lesh, and Carl Francis. I'd especially like to thank everyone, who unfortunately I can't acknowledge by name, who helped out by being a subject in the user trials or during formative evaluation.

I'm very grateful to Lindsay and Dan Palazuelos for both their wonderful work in the field, and for letting me analyse some of the iterations of their protocols and provide them as examples.

I owe everything I know about User Centered Design to the ongoing lessons of Rob Miller, and everyone I worked with in the User Interface Design group. The ever increasing energy and ingenuity of the group was an inspiration.

I'm definitely couldn't have completed this work without the support of my colleagues at Dimagi in being flexible with my scheduling with grace and understanding.

Similarly, I'm deeply grateful to all of my friends and family for all of their ongoing support, which I'm lucky to have always had.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Across many fields of study, from health care to ornithology, there exist experts who regularly make decisions in a structured manner using their extensive knowledge. Decision Support is a way for non-experts use this same knowledge by capturing its structure. Decision Support systems come in a variety of forms, including flowcharts, troubleshooting manuals, and with increasing regularity, computer systems. Most commonly the creation of electronic decision support systems requires at the very least both an expert in the field that the system is being developed for, as well as an expert in software development. This necessity greatly increases the cost of creating decision support systems, and makes iterating and experimenting with particular protocols often infeasible.

1.1 Decision Support

Decision support systems have the promising potential not to replace experts, but to allow their expertise to be used by many. There are a number of simple decision support systems in use every day. Financial firms often work with simple decision support metrics to deal with rapidly incoming data. Many forms used in inspections or surveys include minor decision support logic. There are even a few examples of dramatically successful protocols, such as one used to triage patients at Chicago's Cook County Hospital that achieved both higher success rates and quicker diagnosis than expert doctors[13]. The success of decision support systems has been highly limited, however, by the realities of increasing scale and complexity.

Decision Support systems often focus on machine learning or rules based logic. While these systems provide decision support possibilities at a very large scale of inputs and destinations, less attention has been paid to developing software systems for handling smaller-scale human authored protocols. Although their scale is limited, there are a number of inherent advantages to human authored protocols. There is an inherent trust in protocols which have been created by an expert individual, and their simplicity allows them to be implemented flexibly without much infrastructure or overhead.

1.1.1 Example Protocols

I interviewed a number of individuals regarding forms of decision support logic that they were familiar with or been exposed to. Human authored protocols with similar scopes often came in a wide variety of forms.

1. The IMCI Clinical Guidelines - The Integrated Management of Childhood Illness (IMCI) Clinical Guidelines are a standard of care developed by UNICEF and the World Health Organization to treat common causes of death and disease among children less than 5 years old. It is a fairly large protocol, containing hundreds of questions, and is generally presented in the form of a bound physical book which guides a clinical worker through the protocol. In addition to the paper system, IMCI has been shown to be effective when used as an electronic application[3].

2. OSHA Inspection Guidelines - The U.S. Occupational Safety and Health Administration provides a number of inspection guideline protocols that evaluate the safety of specific environments. An example guideline is the set of rules regarding Fire Extinguisher safety and inspection. These guidelines are presented in a number of forms including plain text and checklists with verbal logic.

3. Cook County Triage Protocol - Chicago's Cook County hospital uses a Clinical Guideline protocol to determine the severity of a patient's cardiac episode and direct them to the appropriate ward[13]. This protocol takes the form of a flowchart diagram that doctors are trained to use, located on the walls of exam rooms.

4. Casual Quiz Programs - There are a wide variety of internet applications intended for entertainment which have a similar underlying quiz structure. These applications involve answering a series of questions and then being provided some ranking or classification. These programs exhibit very similar interactions and representations as more serious classification programs, including their user interface and underlying representation.

1.2 End User Development

End User Development systems have the capability to greatly increase the acessibility of electronic decision support systems by making them more cost effective to implement. However, examples of broadly useful End User Development systems are rare, which is to be expected. Otherwise there wouldn't be a need for highly trained software developers at all. The most successful instances of EUD software are seen in those systems that have a clear understanding of the structure and context of the software that will be produced with it. Software like Microsoft Excel successfully allows users to carry out fairly complicated data centered programming, and even manage flow-control with advanced options, because of attention paid to the end goal of the software and to the context in which is it being used.

Decision support provides an exciting opportunity for creating a powerful end user development environment. The information in a decision support protocol (defined by questions, their possible answers, and the links between them) is structured by its nature. This information is similar to the inputs into an Excel program, making the structure of the end products fairly well defined. There exist intuitive visual metaphors in the form of flow charts and similar diagrams to help users visualize their software, giving non-computer experts a comprehensible way to perceive their software. Additionally, developers of these protocols have most often already conceptualized their programs in the structured and logical manner that will end up defining their protocol, like the lists of instructions or *keys* used to identify animal species[7]. In many cases, these decision support systems are essentially authored outside of any existing environment, giving an excellent context for an end user development environment and enabling the success of such an authoring tool.

1.3 Approach

Decision support protocols are a good encapsulation of expert knowledge for a number of fields. They are used in a limited number of circumstances due to the large cost of creating either a physical representation of what can be a very complex system or of creating an electronic representation that might require outside software expertiese. An end user development environment for authoring an electronic representation of such protocols provides a quick and cost effective way for experts to both describe a protocol and to export its structure to existing infrastructure for executing it.

1.3.1 Design Principles

The most important principle for designing this authoring system is that the system must be based around its user interface and decoupled from an underlying representation. Many field experts have little or no computer programming knowledge, so a well designed user interface is vital for its success at allowing the end user programming task to be completed. Although the generated protocol can be exported into many possible underlying formats, the scope of human comprehensible protocols doesn't change drastically depending on those representations.

Next, the system should be focused on representing protocols that are comprehensible in scale by human beings. There are a number of systems in existence which attempt to tackle the task of creating rules based systems or expert learning systems which infer their own weights and rules. These tasks are fundamentally different, however, from human-authored protocols which are inherently linear and more limited. By focusing on protocols that can be written by individuals, the system can succeed at fulfilling a large niche of tasks which can be automated, but don't have a good existing system to do so electronically.

One consequence of this principle is that the scale of protocols which are expected to be used by protocol developers is at most a few hundred questions. This covers a large amount of existing protocols easily, and covers some of the larger protocols like IMCI at its far end. Protocols much larger than the IMCI protocol are a rarity due to how difficult it is to manage something much larger, so it is expected that no reasonable protocol will be substantially bigger.

Additionally, the issues of increasing scale and complexity that make generating larger protocols difficult should be addressed using common programming techniques whenever possible. The paradigms of modularity and object oriented design have achieved success in managing scale in other programming systems, and they should be exploited in this development environment.

The final principle is that the system's development should be oriented on providing a functional, rather than precise, experience. Developing with more complex languages requires a conceptual understanding of many precise concepts. For instance, object oriented languages require the developer to have a mental model of the concept of differentiating between the definition of a class and its many potential instances, which is a common stumbling block for new programmers. Since the real world analogues of the electronic protocols that will be developed contain none of these concepts, the user interface should provide a similarly functional experience. Although this limits the scope of protocols, it falls within the guideline of representing human comprehensible ones.

1.4 Contribution

In order to improve the ability of end-users to design electronic protocols this thesis presents a prototype end user development environment called Scientia. This system provides a way for protocol authors to create a protocol in a format that is portable and can be exported to engines which can execute the protocol which authors have designed. The environment was created following user centered design for users who previously would not have been able to use an electronic environment to execute their protocols without a software expert translating their protocol into an electronic format.

This research builds on a body of design which has gone into the representation of electronic protocols in a portable and standardized format. The goal of this environment, however, is to provide the proper environment to allow authors who do not have the expertise to create documents which can be translated into these formats, rather than to provide the specific semantics of the protocol representation language itself.

In this thesis, we provide a description of the process that was followed to design Scientia's user interface, as well as describing the techniques used to assist authors in creating protocols and an evaluation of their effectiveness.

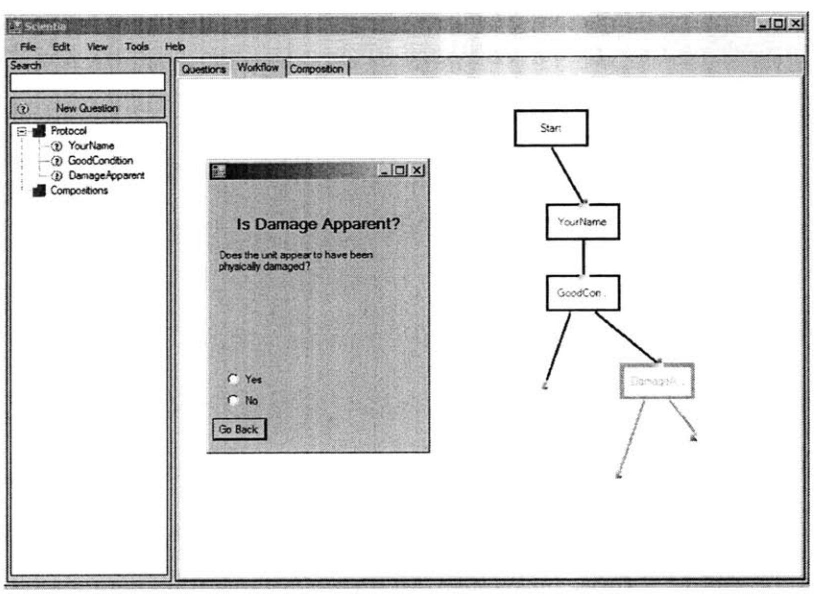


Figure 1-1: A protocol being authored and tested in Scientia

1.5 Evaluation

Scientia's interface was evaluated with the help of individuals who were familiar with the authoring of protocols outside of standard software development languages. A sample of these individuals were asked to design and implement a protocol within the interface, including steps for testing that protocol for correctness. The system was tested for its ability to help its users create full protocols that could be used electronically, for its ease of use and level of assisting protocol authors to accomplish that task, and for the success of the individual techniques used to combat the issues identified as problems with creating electronic protocols.

1.6 Contents

The remainder of this thesis is organized in the following manner.

Chapter 2 discusses the state of authoring tools for electronic protocols, as well as related work on the use and value of decision support protocols when used on eletronic devices.

Chapter 3 review the process of problem evaluation that was undertaken to identify the difficulties arising from authoring protocols in an electronic format. It provides a sample of some decision support protocols which have been used, along with an evaluation of what difference between electronic and non-electronic protocols are and what burdens the level detail required for electronic formats places on authors.

Chapter 4 presents a detailed description of Scientia's user interface. This chapter describes the ways in which the interface is configured to address the difficulties established in the problem analysis in a way which provides authors the level of specificity necessary in their protocols, without being overwhelmed by the complexity of designing detailed and potentially complicated machine logic.

Chapter 5 presents the architecture of the system along with the specifics of how the techniques describe in the user interface were implemented in the software.

Chapter 6 describes the study conducted to evaluate the efficacy of the techniques used to make designing electronic protocols more accessible.

Chapter 7 offers conclusions from the thesis work, and suggests further work to be conducted towards making the design and implementation of electronic protocols feasible and simple for end users.

Chapter 2

Related Work

2.1 Form Builders

Commercial 'Form Builder' software has often included decision support mechanisms for building simple applications. One notable modern system is the Pendragon Form Builder. The Pendragon authoring environment enables users to capture form data per screen, and to branch the next screen based on prior answers. The system has been successfully used by non-computer experts to create multiple examples of decision support systems, as seen in a University of Alabama public health system designed to counsel and refer patients based on decision rules[2]. It is widely used in commercial settings to improve automation for tasks like the inspection of components. Workflow logic in these systems is typically costly in design time for users because the rules designed are most often for skipping over or enabling questions individually, see Figure 2-1. Since the rules are at such a low level and require the user to often write logic by hand, it is difficult to use to obtain complex decision protocols.

2.2 Process Automation

The automation of different processes, especially in the cases of emergency management, medical tasks, or high-risk industrial inspections raises a number of questions regarding to methodology and safety. The safety of using computers in high-risk

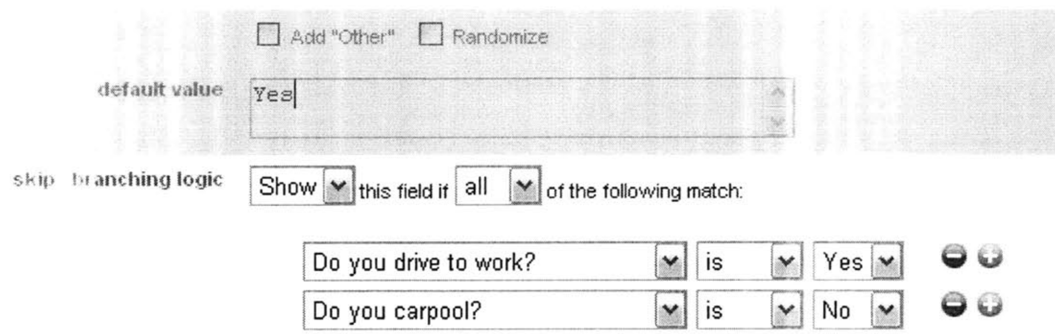


Figure 2-1: An Example Screenshot from the FormSpring Commercial Form Builder

fields has been the subject of scrutiny in the past. Results have been positive for devices and software that are designed with safety in mind. One study evaluated a user interface for a mobile device used to help coordinate the movements of a large container ship via radio. This study concluded that the device would actually increase safety in the process[5]. Another evaluation of computer systems for the use of distributing prescriptions found that usability problems do have the potential for creating errors[6]. This implies that extensive field testing and usability studies regarding specific interfaces are necessary. The reuse of decision support protocols in the same client software will allow these safety benefits to be applied to multiple applications.

2.3 End User Development Environments

End User Development environments in the fields of expert and knowledge based systems have been a field of study for nearly two decades of academic research. The Protege system, a knowledge-based expert system developed by a small academic community, has achieved a large degree of success at creating a development environment where it is possible for users to classify rules, and to build software on top of the knowledge modeled by their artificial intelligence system[4]. Protege has grown from a specific medical context to a much larger expert system, allowing the classification of more complicated rules and the ability to integrate with other systems. The user interface for generating ontologies in Protoge is strongly focussed on creating data structure, rather than interaction, making it of limited utility for creating protocols. Ontologies authored using Protege are usable in a few existing systems, but largely as structure for data in those systems, as the knowledge bases created in Protege are not commonly sufficient to define a protocol.

2.4 Protocol Representation

There has also been a great deal of work in designing representation languages for decision protocols. One notable effort to represent decision protocols in a computer-interpretable format has been the GuideLine Interchange Format (GLIF) developed in the medical community by the InterMed Collaboratory[9]. GLIF is a second generation protocol specification format, created by integrating the experience from the development of four different guideline representations. GLIF has achieved a large level of success in the representation of diverse medical guidelines, and has potential for a standardized internal representation of protocols[9]. Columbia University is currently implementing GLIF as an aspect of an existing computer-based physician order entry (CPOE) system to enhance the system's decision support capabilities[1]. The protocols supported by systems like GLIF are extensive, and are an excellent starting point for encapsulating the control flow logic of the kind of decision support protocols I strive to create.

2.5 Rule Based Expert Systems

Research in the artificial intelligence community has long realized the value of rule based systems, although the focus of these systems is generally to intelligently generate the rules. These *expert systems* are artificial intelligence systems generally designed to exceed human capabilities by accepting a specific set of information and making new rule-based assessments on incoming data in a narrow range of results. Early examples of expert systems include the Mycin, a 1970's Stanford system for recommending a proper antibiotic and dosage for blood-borne infections[14]. Mycin's results showed it to most often outmatch all doctors but those specially trained in blood infections. Although my proposed system is designed to have users create these rules, as opposed to artificial intelligence algorithms, the observation that computer systems can follow protocols to achieve results that would be delivered by, or even outperform, expert human actors is extremely relevant for the success of any decision support software system.

Chapter 3

Problem Analysis

Paper based decision support protocols are fairly common tools utilized in the fields of medicine, industry, and public health. The previously discussed examples of protocols for identifying disease and performing safety inspections on equipment demonstrate the utility of encapsulating these processes and knowledge in the real world. However, there are far fewer examples of users successfully using rich electronic decision support tools.

Utilizing electronic devices like computers, phones, and PDA's, requires an author to create a protocol in a more rigid format than paper protocols, which can outline steps to be followed in a general and unspecific way. Although there currently exist tools to assist authors in creating specific decision logic, including low-level programming environments like Microsoft Excel, they suffer from a number of problems that make the tools difficult to use for authoring protocols of the scope that authors want and need.

Creating an environment that can be used to create protocols requires more than the ability to take an existing protocol and make it electronic, since the representation of the protocols can be so radically different. In the attempt to create a tool that would allow authors to be the most succesful at designing electronic protocols, I evaluated the design of these protocols both on paper and electronically to determine where the difficulties lie in translating between those formats, so that these issues could be addressed by effective user interface design.

3.1 Methods

I obtained a number of different protocol representations that were created by individuals who were experts in the field that the protocol was being used for, but who were not experts in computer programming. Any non-publicly published protocols are made available in Appendix B. The formats of the protocols presented included flowchart-style diagrams, Microsoft Excel spreadsheets which specified logic, plain English step by step instructions, and protocols defined in form creation tools. Some of the most complicated protocols were represented with mixed formats. Complex public health protocols often resemble flowcharts, but contain a good deal of plain-English instructions, as illustrated by Figure 3-1.

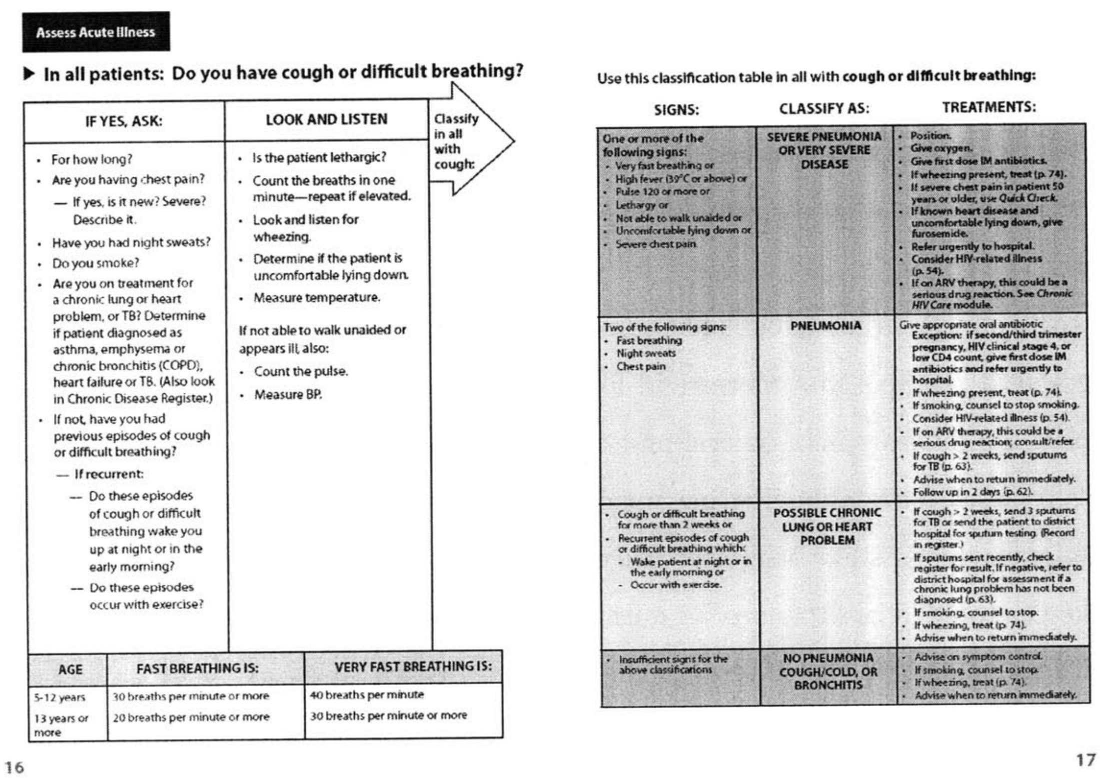


Figure 3-1: The WHO's IMAI Protocol is a mix of Flowchart and English Instructions

I was also able to discuss the creation of many of these protocols with their authors in informal interviews. I inquired into both what difficulties they generally encountered in authoring the protocols they were creating as well as what kinds of mistakes were common in the protocols created.

The following major protocols were studied in-depth to analyze the field of protocol creation. Some other short examples were studied in less depth, such as a museum object acquisition protocol from a textbook.

1. IMCI - The WHO's Integrated Management of Childhood Illness[16] protocol is a standard for treatment of chidren under five years old. IMCI is utilized in more than 75 countries around the world, and is a common candidate for making into an electronic protocol. IMCI was analysed in its booklet representation as well as in other intermediate and electronic forms[3].

2. IMAI - The WHO's IMAI[15] protocol is a similar standard for treatment of adults for common causes of mortality. IMAI was analysed in its booklet representation.

3. Adult[10] and Children's[11] Diarrhea Protocol - These protocols are used to handle the treatment of acute and chronic dehydration among children and adults, initially authored for evaluation in a clinical setting in Central America. These protocols were analysed in flowchart, paper, and electronic formats.

4. Epilepsy Management Protocol[12] - This protocol is designed to help assist the management of epilepsy in a clinical setting. Initially authored for evaluation in a clinical setting in Central America. These protocols were analysed in flowchart, paper, and electronic formats.

5. e-CTC Patient Assessment Protocol[8] - The e-CTC protocol was designed to improve the ability of clinicians to manage resources by providing questions and an outcome assessment of patients which can triage them according to need. The protocol was analysed in a paper format.

6. WHO Pregnancy Essential Practice Guide[17] - This protocol is used to guide the assessment and care of pregnant women inside and outside of a clinical setting. This protocol was analysed in its paper format.

Using the samples I performed a qualitative analysis of the aspects of the protocols that could easily be transferred into logical statements, as well as which aspects of the protocols required human intervention to clarify logical reasoning or disambiguate a branch of the logic.

For one of the sample protocols, multiple iterations were available as the authors attempted to refine from a general paper protocol to more specific logic statements for an eventual electronic protocol. This protocol was analyzed to see what aspects of electronic protocols were easy or intuitive for designers to capture from their protocol, and which aspects were not successfully adapted to an electronic format. The protocol was initially in the form of a flow chart, and the final iterations were provided in a spreadsheet step-logic format.

After this qualitative analysis, I determined three key major sources of potential difficulty that arose in translating the intuitive human paper format into electronic protocols. Addressing these difficulties guided the format and design of the user interface for the authoring tool. In addition, I discovered a number of more minor issues that could be addressed in improving the ease of authoring correct protocols in that tool.

3.2 Major Issues

3.2.1 Specificity

The most common aspect of design that both designers reported when creating protocols, and which was evident from existing protocols was determining the appropriate level of the logic used for navigation through their protocol. Before a protocol can be authored into an electronic system, and often before it can be successfully used in any format, it must be wholly well-formed and unambiguous. The issue of specificity is how much of the protocol is represented in a format that can not be clearly expressed as logical statements. Flowchart representations, like the one in 3-2, often cannot be decomposed into logic unambiguously.

This issue was extremely apparent in the sample in which the protocol was iterated on. Achieving a completely unambiguous format stuck out as one of the major difficulties in that process. None of the formats of the sample protocols was wholly best in avoiding ambiguity, with each suffering from somewhat different ambiguous representations.

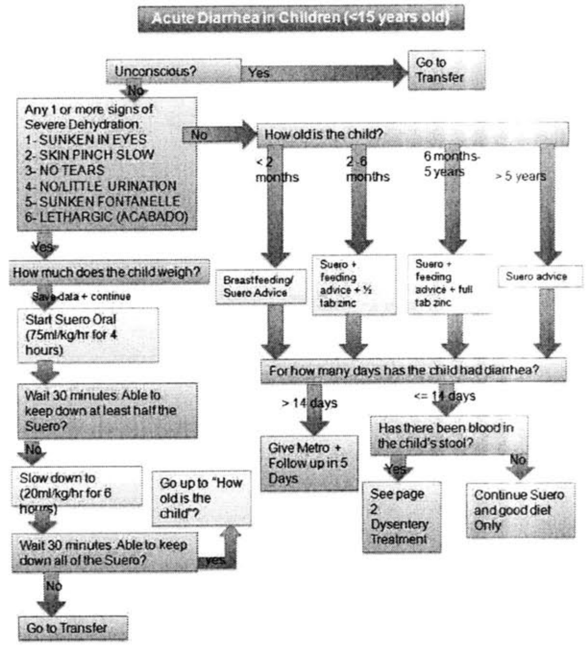


Figure 3-2: An example of a protocol represented as a flowchart

The flowchart diagrams commonly contained paths which fell back to some human-interpreted direction in order to proceed. In Figure 3-3, not all combinations of outcomes are represented by the flowchart and some outcomes provide transitions that require breaking the general format of the flowchart's transitions. For instance, the path that reads "Follow instructions for..." is easy to follow for a human, but not directly expressed logically. This kind of path often requires that the protocol be followed by someone capable of determining the logic rule being used and then whether it applies, like the "Yes" response to "Any 1 or more of..." path in Figure 3-2.

An example of one of the samples in a spreadsheet format is given in Figure 3-4. These protocols have each decision or question enumerated, and specify either relevancy conditions(Figure 3-4) or Go To instructions(Figure 3-7) for where to proceed. The transition for each branch is most often unambiguous since the outcomes are always directives to a specific point in the spreadsheet. However, the lack of any direct visible correlation between each section of the protocol and the next means that transitions may miss sections of questions which are expected to be asked. It is much more difficult to verify that each possible branch is represented due to the lack of any visual cues about each transition.

For many of the protocols a certain lack of specificity was expected and often

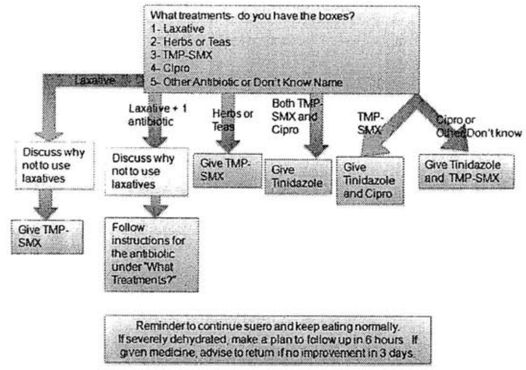


Figure 3-3: A branch easily followed by a human, but containing ambiguity

reasonable since the protocols were mean to be interpreted by humans with some level of expertise in a subject. These ambiguities were meant to be resolved by a human with domain-specific knowledge, for instance a protocol asking whether a fire extinguisher is adequately charged relies on some external source knowing how to identify the state of the extinguisher and whether that state is within the boundaries requested. This kind of ambiguity is a valid feature of many protocols, and doesn't contribute to the protocol not being well-formed.

WEIGHT			
	yes	*Have you lost weight since your last visit? If yes, how much?*	
	No	Have you lost weight on your face, arms, legs, or buttocks?	no; yes
prev=yes	Yes	Does this weight loss bother you?	no; yes
	No	Have you gained weight on your abdomen, breasts, or back of your neck?	no; yes
prev=yes	No	Are you gaining more weight there than in other parts of your body?	no; yes
prev=yes	Yes	Does this weight gain bother you?	no; yes
Opportunistic Infections (TB & Coughing)			
	no	Are you on TB treatment?	no; yes
prev=yes	Yes, if NOT feeling better	Are you feeling better on TB treatment?	Feeling better; Not feeling better
tb=no	no	Are you coughing?	no; yes
prev=yes	Yes	Does anyone in your home have a cough or TB?	no; yes
cough=yes	Yes	How long have you been coughing?	Less than 2 weeks; Longer than 2 weeks
cough=yes	yes	Do you have bloody sputum?	no; yes
prev2=no	No	Advise: return to clinic if you continue to cough for more than 2 weeks or cough up blood	OK
	no	Are you sweating at night?	no; yes
prev=yes	no	Does the sweat soak your bedding and bed clothes?	no; yes
prev=yes	yes	Does it happen every night?	no; yes
	no	Do you get short of breath when you walk around inside your house?	no; yes
prev=yes	yes	Is it new or getting worse since your last visit?	no; yes
General OIs section			
	yes	Do you have pain or sores in your mouth or trouble swallowing?	no; yes
	no	Do you have stomach pain or discomfort?	no; yes
prev=yes	yes	Is it new or getting worse since your last visit?	no; yes

Figure 3-4: An example protocol represented as a spreadsheet

Electronic protocols exhibiting this well-formed ambiguity are more error prone, though, as they could be evaluating the statements directly. Returning to the example of whether a fire extinguisher is charged, the protocol could instead ask for the internal pressure of the extinguisher and then provide the appropriate decision directly based on evidence, which may lead to less errors. Encouraging this use of an electronic protocol's abilities would be a positive aspect of an authoring tool.

3.2.2 Scale

Effectively managing the scale of hundreds of questions and the possible paths of traversing them through a protocol was a difficulty that was evident even in protocols in plain text, which provide the most freedom of representation for the potentially complex rules involved. Comparing between the size of the different protocols leads to interesting inferences about how protocols are designed.

One important aspect of larger protocols is that none of the largest protocols (IMCI and IMAI) analyzed were flowcharts or highly specific. Both of these protocols are a hybrid of a flowchart and a plain-text set of instructions, placing a set of plain-text instructions into a larger flowchart-style diagram like that in Figure 3-5. This implies that designers have some mental mechanisms available for dealing with very large protocols that involve encapsulating groups of questions. Unfortunately, this also means that the largest protools available are in a format which is difficult to transition directly into logical statements.

The general trend was for shorter protocols to be mostly specified in fairly specific formats with some sections which are incompatible with strict rules, and for the larger protocols to be more reliant on a fairly vague format for individual question transitions, but follow a fairly explicit path through deliniated sections from beginning to end. Additionally, the larger protocols generally followed a more linear path when compared to the smaller protocols, which occasionally had logic that required looping or individual transitions that were extremely complex, like the "Follow instructions..." outcome described earlier in 3-3.

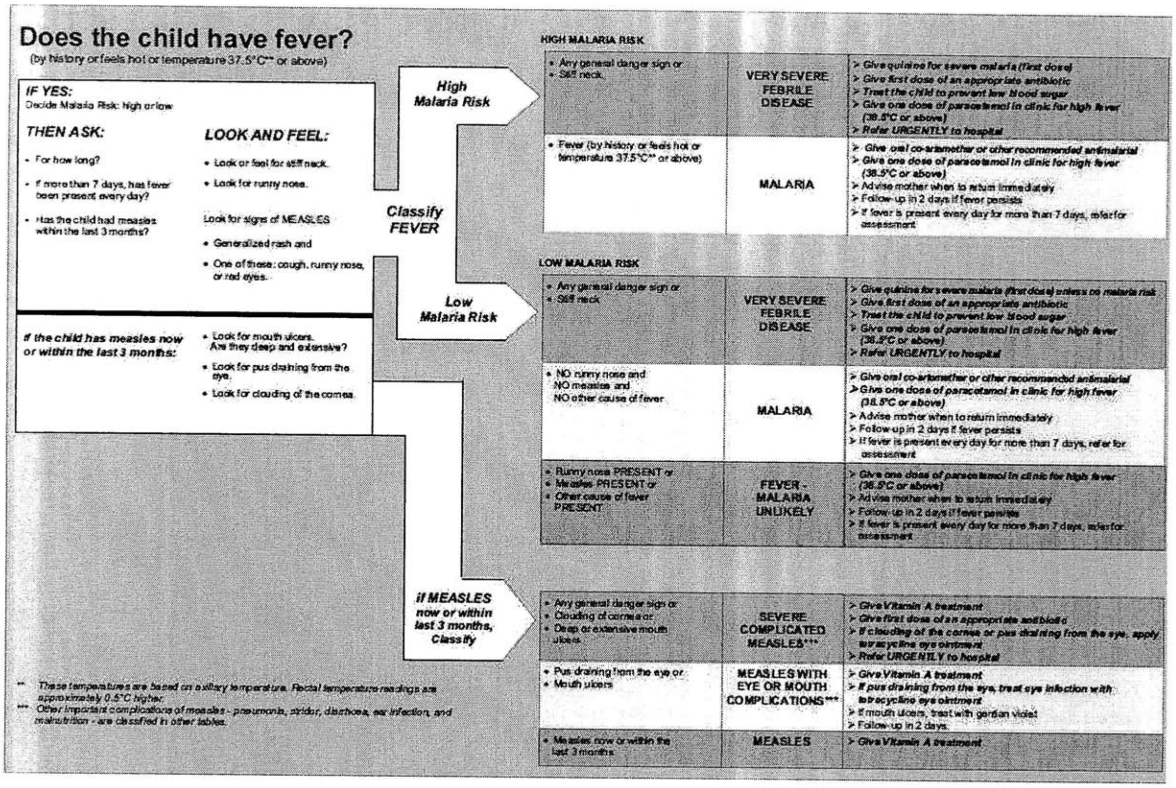


Figure 3-5: The organization of the IMCI protocol booklet is highly modular

3.2.3 Complexity

As the scale of protocols increases in terms of numbers of questions, the number of strictly unambiguous logic rules needed to represent that protocol often increases dramatically. Especially for questions or decisions with a large branching factor, representing all of the different possible branches becomes a serious concern. The problem of complexity is distinct from that of specificity in that while specificity is an issue that requires authors to understand and define potentially difficult ways of expressing a decision, like defining the necessary information to describe "Laxative + 1 antibiotic" as a decision, complexity deals with the difficulty of managing the resulting expansion of ideas.

A concrete example of a situation where translating a seemingly straightforward situation on paper into explicit decision logic can be seen from the sample epilepsy dosage protocol in Figure 3-6. The intent of this section of the protocol is to potentially ask three different sets of questions depending on whether a patient is taking any of three different medications. The initial protocol on paper is able to easily describe the instruction to ask any of the appropriate questions, but creating a fully unambiguous version of the protocol requires a great deal of complexity to be introduced.

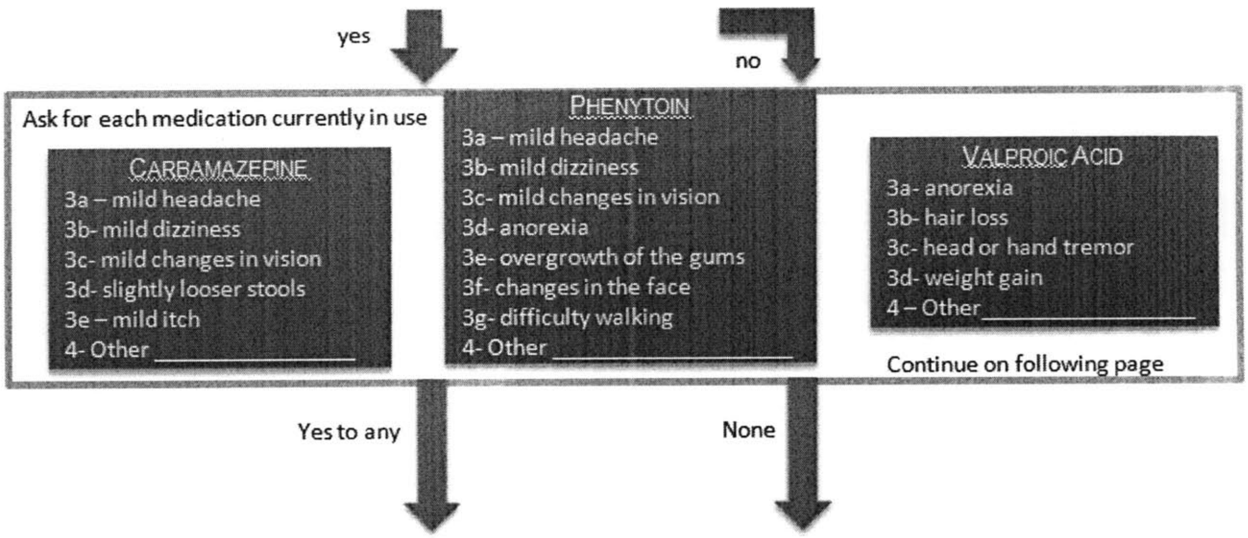


Figure 3-6: A a straightforward flowchart construct from an epilepsy protocol

The first source of complexity is in the expansion of possible transitions resulting from the initial question of what medications are active (essentially what further decisions should be visited while executing the protocol). In this case, the increase in complexity can be very, very large since the protocol could potentially split into up to three different branches. This is not a necessary outcome of the specification, but occasionally could not be avoided.

Another source of complexity is in the transition from each of these states to the rest of the protocol. Since each state can individually lead to a particular set of further questions being asked, a high branching factor has again been introduced.

The source of difficulty in each of these cases can be identified in the next iteration of the described protocol, which is in a spreadsheet format. The transition rule defined in that representation for each of the exit points as illustrated in Figure 3-7 is given with the instruction "Continue through corresponding Q(1-3) and/or then go to R". In this situation, the complexity is addressed by defining an internal state for the protocol, namely three binary values which are used to perform an internal branch about whether to proceed down a specific path.

Q1	Question to Patient	Check Box	Carbamazepine. Have you been experiencing the following as a side effect? 3a - mild headache 3b- mild dizziness 3c- mild changes in vision 3d- slightly looser stools 3e - mild itch 4- Other ______________	Yes to Any	Continue through corresponding Q2-3 and/or then go to R
				None	Continue through corresonding Q2-3 and/or then Go to V
Q2	Question to Patient	Check Box	Phenytoin. Have you been experiencing the following as a side effect? 3a - mild headache 3b- mild dizziness 3c- mild changes in vision 3d- anorexia 3e- overgrowth of the gums 3f- changes in the face 3g- difficulty walking 4- Other ______________	Yes to Any	Continue through corresponding Q3 and/or then go to R
				None	Continue through coresonding Q3 and/or Go to V
Q3	Question to Patient	Check Box	Valproic Acid. Have you been experiencing the following as a side effect? 3a- anorexia 3b- hair loss 3c- head or hand tremor 3d- weight gain 4 - Other______________	Yes to Any	Go to R
				None	Go to V
R	Question to Patient	Check Box	Is this side effect 1- tolerable, you can keep taking your medicine? 2- intolerable, you want to stop the medicine?	1	Go to S1

Figure 3-7: The use of state to avoid drastic increases in number of transitions while codifying Figure 3-6

The use of state to manage complex branching was common across many protocols, including the largest ones. It greatly assists in collapsing branching factors by enabling a step in the protocol to affect a later decision in the protocol. It does, though, introduce the necessity for a place in which to store such state, and a way in which to uniquely reference it. This problem is familiar in a software engineering context, but is at a level of complexity which is undesirable for authors without a software development background. As such, it is desirable to provide the use of state to simplify decisions without requiring the in-depth management of variables or memory that would exist in more complete software.

3.3 Minor Issues

3.3.1 Testing

Even if a protocol in paper format is well-structured, it provides very little feedback about the process that is actually followd when the protocol is used. For instance, a flowchart might have an unambiguous structure, but when used might not actual encapsulate the protocol the author intended. These kinds of errors are analogous to bugs in syntactically well-formed computer code, but are much more difficult for authors to debug.

Chapter 4

Interface Design

Scientia's user interface was designed to provide a straightforward way to interact with the common elements used in the existing protocols created by target protocol authors, namely questions that are asked by the system, and the paths that are followed in the protocol based on each answer.

Based on feedback about general design processes for authoring protocols, the activities for creating questions and editing the workflow paths that define the protocol are performed seperately. Although it is fairly easy to move quickly between editing a protocol's workflow and its questions, the interface is optimized for creating a bank of questions and laying them out afterwards.

4.1 Question Editor

Users define questions that will be asked in the protocol as question objects. Questions are defined by a set of prompting information (title, text, possible audio or visual media) and by their possible outcomes. The two most common types of outcomes are selecting from a list of choices and entering a value.

When creating and editing question objects, the author is presented with a visual representation of how their question might look in a real decision support engine. This look and feel emulates the small screen of a portable electronic device and is also used when testing and debugging a protocol, and will be elaborated on in that

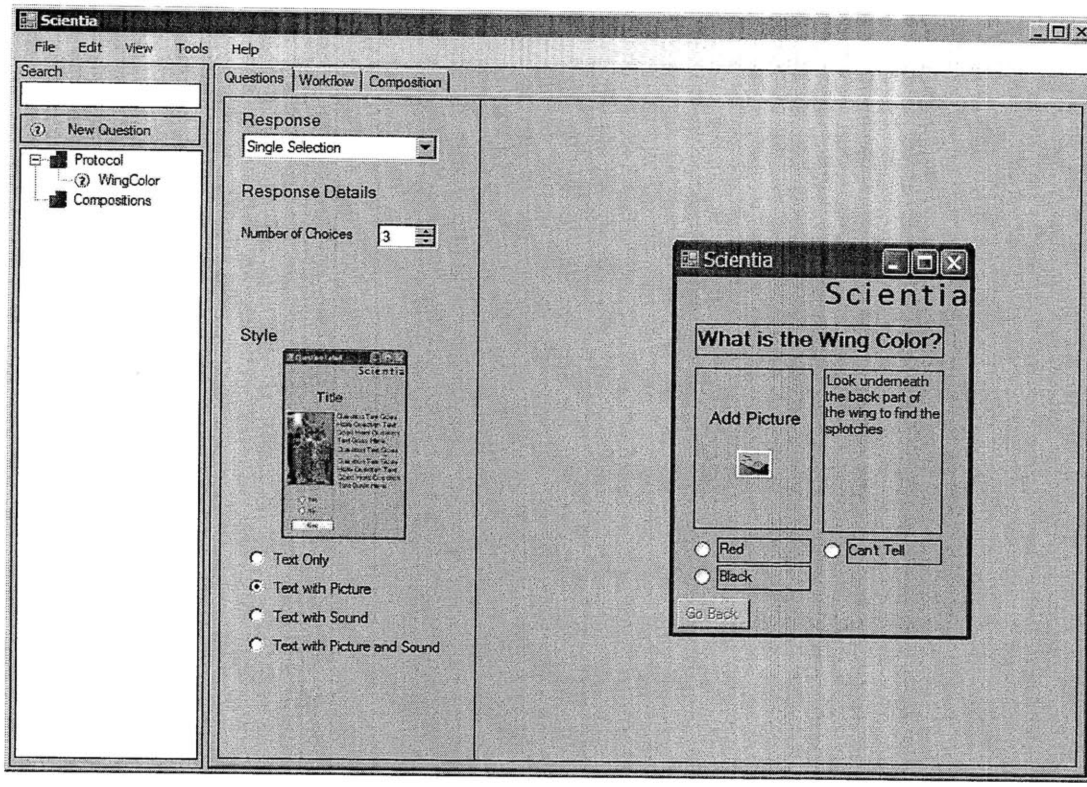


Figure 4-1: Scientia's Question Editor

section.

4.1.1 Question Types

There are three basic types of questions available, which are based on common entry modes for existing protocols.

1. Selection - The client presents a prompt to the author, and provides a list of options for possible responses to that question. In Scientia, the size of the list is fairly limited (between two and four) in order to keep the workflow for questions straightforward and manageable. Handling many options isn't supported in this type of question. When designing a protocol having large numbers of options would be difficult due to the number of resulting edges from that point in the protocol. Additionally, In the user interface in a decision support client, the user interface would be different for choosing between small numbers of options,

which can be selected, and large numbers of options, which might have to be searched, filtered, or tagged.

Selections are the most common form of question response in real world protocols, often with a small number of options such as "Yes" or "No". An example selection question is provided in Figure 4-2.

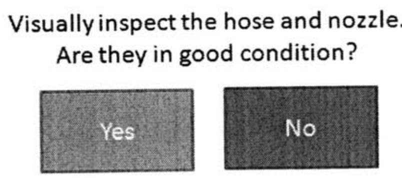


Figure 4-2: A Sample Yes/No Question Adapted From a Fire Extinguisher Inspection Procedure Intended to Meet SORM Chapter 6 subchapter 6.6 and OSHA 1910.157 Guidelines

2. Text Entry - The client presents a prompt to the user, and provides a space for entering either free text or certain restricted types of text as a response, along with a button that allows the user to finish the decision. Scientia allows authors to specify common types of restricted text, or to specify an input mask defining exactly what characters count as a valid response. Masked text is useful for entering data in very specific formats like a social security number or other id number.

   Text Entry is also a common form of response in real world protocols, although its format is extremely variable across platforms. Input validation on paper is impossible, but the range of restrictions is unbounded. The interface only provides high-level restrictions of "Whole Number", "Any Number", "Free Text", and "Masked Text" in order to maintain the goal of simplicity, to emulate the real world, and not overwhelm users with generally unnecessary options.

3. Informational - The client presents a screen to the user which provides information, along with a button to signal that they have read the text.

   The purpose of the prompt is to provide information to the user without recording any kind of response. The nomenclature of including this as a "Question Type" was supported by experiences with authors familiar with protocols, who reported that the concept was clear and familiar.

It is worth noting that there are other possible question types which exist in certain protocols and not currently supported in Scientia such as GPS Location, Date, Multiple Selection, and integration with electronic data sources like a camera, thermometer, or other digital sensor. Appropriate UI's for specifying these kinds of questions is left for future work.

4.1.2 Media Support

Many protocols include not only text in their prompts, but also different kinds of media to assist the user in answering a question. Species classification protocols, for instance, make heavy use of images and occasionally sound when prompting the user about certain features.

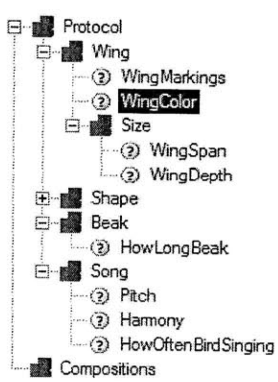


Figure 4-3: Questions placed into groups in the explorer

Scientia allows images and sounds to be included in a question by clicking the appropriate prompt and selecting the associated media. Like the rest of the question editor, this visual representation is consistent with the debugging environment.

4.2 Element Explorer

The left side of the user interface is dedicated to a list of elements that have been defined by the protocol author, along with tools for managing them. The explorer contains the questions and subprotocols that the author has defined in a hierarchical list. It also provides an interface for searching for a particular question via incremental search.

4.2.1 Grouping

Protocols that are on the medium to small size (around 15 or 20 questions) don't require any special consideration for scanning the list of questions and locating the appopriate one. When handling larger protocols, however, it can be cumbersome to scroll through a large list of questions to locate one. The element explorer shown in Figure 4-3 allows authors to manually create collapsible nested groups in this list and thus organize questions.

4.2.2 Search

The explorer also allows for questions to be searched via an incremental search. When text is entered into the explorer's search bar, all questions containing that text are made visible by having their parent groups expanded, and are made distinct with yellow highlighting.

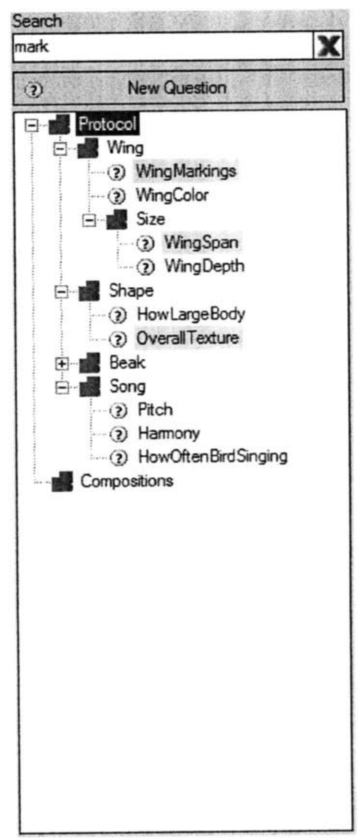


Figure 4-4: Performing a search for text inside of created questions

4.2.3 Element Interaction

The behavior of selecting elements in the explorer varies slightly from screen to screen depending on what the most intuitive interaction with that element is at the time. While in the question editing mode, selecting a question displays that question on the screen and allows it to be edited. Selecting a question while in workflow editing mode, however, does not result in the same behavior but rather highlights any instances of that question in the workflow diagram.

Although this behavior is not perfectly consistent, authors found it intuitive and remarked that they expected the interface to react to their selection of each element on the current screen.

4.3 Question Identifiers

One common complication identified in authoring large protocols was the problem of generating absolute references for each of the potentially hundreds of questions that were part of that protocol. In the protocols that were reviewed when designing Scientia's interface, a number of different protocols were in a format requiring questions to have identifiers, and in all of these cases they were either assigned fairly cryptic unique identifiers for each of their questions such as "ChmtSparesBesties", "A", "1.C.3", or the questions were referred to vaguely in a non-unique manner.

4.3.1 Motivation

There are a few reasons why unique identifiers for Questions are difficult to write. One is that authors tend to write identifiers based on the scope of the questions around the current one. This makes the question easily discernable from its immediate neighbors, but difficult to identify when context is lost later. An example of a common pattern is to be asking a set of questions related to the type of a particular animal's spots, naming an initial set of questions "malespotlocation" and "malespotcolor", but eventually tiring and simply naming further questions "shape", "size", assuming that the existing context will always make those questions sensible.

Another common problem was disambiguating questions once an old identifier was no longer sufficient. Many of these situations resulted in unique ids being duplicated, but disambiguated by adding numbers afterward or other arbitrary suffixes. This causes serious recognition and discernibility problems later, as both values tend to now be difficult to discern.

4.3.2 Solution

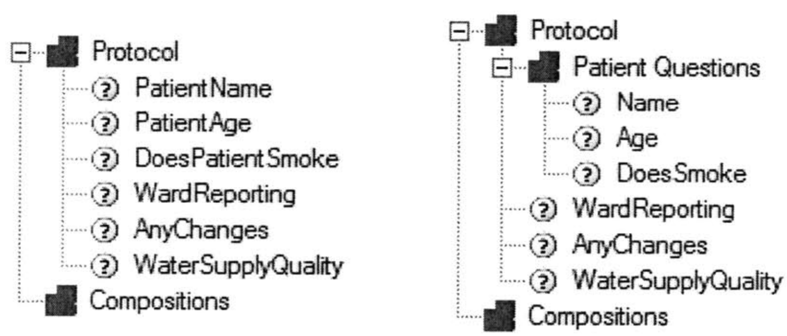
In order to prevent this form of user error, Scientia does not prompt authors for an ID to be assigned to individual questions, but rather generates an identifier for the author based on the features of the question. Furthermore, the identifier for a question is not fixed. The id is updated if a better identifier is established later.

This identifier has the advantage that it reflects the question's content, requires less work from the author, and doesn't require the author to edit multiple fields when changing the content of a question.

4.3.3 Construction

Identifiers for questions are constructed using data from a question's title, after removing whitespace, punctuation, common words and determiner words from the text. For instance, a question with the title "What is the Patient's Age?" would receive the identifier "PatientAge" as an initial step, and would dynamically receive the identifier "Age" if the word "Patient" was present in the majority of words in the heirarchy due to another question being changed. The algorithm is described in some further detain in Section 5.2.

This commonality is reflected not just in the full list of questions, but in each relative grouping as well. For instance, if there were a number of questions which contained the word "Patient" mixed into a flat set of many questions, their ids would contain the word "Patient". If those questions were all in one group, however, they probably would not due to the relative frequency of the word in that grouping. An example of this is provided in Figure 4-5.



(a) Identifiers in one flat group

(b) Identifiers in a group have words removed that are sufficiently common

Figure 4-5: Example of Common Word Relativity when Building Identifiers

4.3.4 Alternatives

There are a number of disadvantages to this dynamic identifier scheme due to the serious violation of consistency when an identifier changes. One alternative would be to generate identifiers upon a question's creation and then to commit to those ids, another would be to provide this question identifying capability as an action that the author can perform, but to allow them to specify the unique id otherwise.

Dynamically generating ids is a preferable approach to these alternatives since the goal of the id is to provide the author the ability to recognize a question using its id based on their knowledge of the protocol's content, rather than any new information. Experienced authors are extremely familiar with the content of individual questions, and can already identify which question is being referenced by a small amount of information about it. This scheme focuses on providing the most relevant portion of a question's content to the author in order to identify it based on its content, rather than by forcing them to summarize it into a unique identifier manually, which they may not do well. By leveraging the author's knowledge of the protocol in advance, the need for this summarization can be avoided.

4.4 Workflow Editor

The transitions that define which questions are asked after others are answered is defined by a protocol's workflow. The workflow editor is a direct manipulation canvas which provides a visual layout of the structure of the workflow, and allows the author to build and edit that layout.

The workflow is represented by sets of blocks, each of which represent a step in the protocol. Blocks represent different units of computation. Asking a question is the most common unit, but each of them have the same structure in the workflow. Blocks have a single entry point, which can be reached from any number of other blocks, and have N possible exit points which can lead to other blocks, depending on their content. For instance, a block representing a question with 3 choices will have 3 exit points.

The points of entry and exit from blocks are represented visually in the canvas as edges which protrude from a block's top and bottom. At the end of these edges is a small node which can be dragged around the canvas to manipulate it.

Users create a workflow protocol by linking together block inputs with block outputs in order to establish paths through questions. The workflow has a singleton root element which serves as the anchor for where the protocol begins. The end of the protocol is unspecified. In order to reduce the amount of effort on the author for the massive branching factor at the end of a protocol, blocks which have exit points that are unassigned will simply act as a terminus.

4.4.1 Types of Blocks

There are a few different types of blocks that authors will interact with when creating protocols.

- Root Block - The root block represents the entry point into the protocol. This block is always present on any workflow or subworkflow, and cannot be deleted. It always has no entrance points, and exactly one exit point.

- Question Block - A question block is used to add a question to the protocol's workflow. When a protocol is executed and the workflow enters the question block, the particular question that has been designed by the author is asked. The answer to that question determines which exit is taken while proceeding in the protocol. The number of exit points from a question block is defined by the number of answers specified in the Question creator.

- Subprotocol Block - A subprotocol block represents a composition of workflow logic. When a protocol is executed and the workflow enters a subprotocol block, the subprotocol is executed like the overall protocol, starting at its root. Any exit point for a block in the subprotocol which is not connected to another block in that subprotocol will propagate downwards to be an exit point of a subprotocol block as illustrated in Figure 4-6.

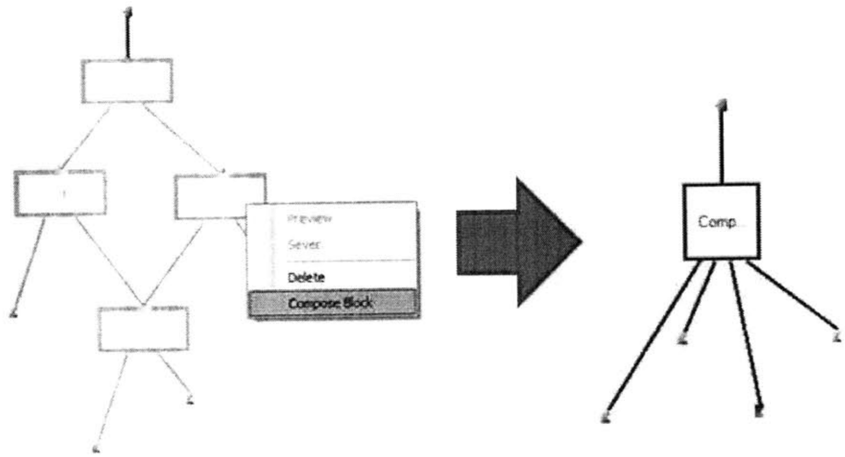


Figure 4-6: Composing a sub protocol and the resulting block

4.4.2 Workflow Actions

The workflow interface supports a number of direct manipulation actions that are used to build protocols.

- Creating Blocks - Blocks are created by dragging and dropping different questions or compositions from the Explorer onto the workflow canvas. There can be many blocks representing the same question, so the author isn't prevented from dragging questions into the workflow multiple times.
- Connecting Blocks - Two blocks can be connected together with an edge by dragging together the potential edges that protrude from them. Another techique would have been to allow edges to be created on their own as seperate entities to manually create connections, rather than just manipuating them. This approach was chosen to reinforce information about the block (number of exits), and to provide a visual cue for how connections can be made.
- Breaking Connections - When the author mouses over a connection, a razor blade icon pops up over the connection's center. Clicking on the

blade cuts the connection, and each block regains its initial protruding edge to be reconnected.

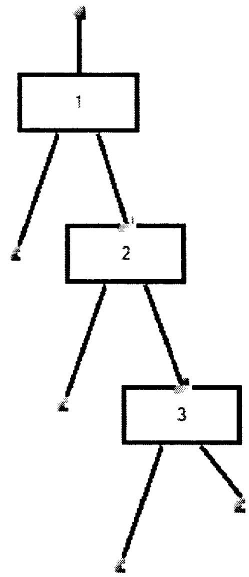
- Selecting Blocks - Certain actions, like composition, take place after selecting multiple blocks. Blocks can be selected in some of the common ways that elements are selected in direct manipulation interfaces. Shift-clicking a block will add it to the current selection, and a selection rectangle can created by holding down shift while clicking and dragging to create a marquee. Since the default mode for dragging the canvas is to pan, shift was selected for its similarity when selecting multiple blocks.

- Compose Subprotocol - When multiple blocks are selected, their current state of connections can be composed into a subprotocol, which can then represent those questions as a unit later. In order to compose a subprotocol, a number of blocks should be selected. The context, or right click, menu that appears for any of those blocks will now allow for the current selection to be composed.

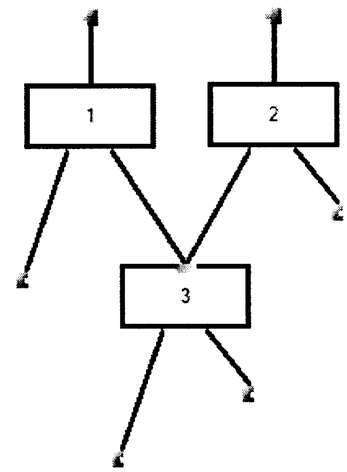
  Some selections of elements may not make for a valid subprotocol. Subprotocols must contain only one point of entry, meaning that only one block in the selection can be connected at its head to either a block outside of the protocol or no block at all. For instance, in a protocol with block 1 connected to block 2, and block 2 connected to block 3, blocks 1 and 3 could not compose a valid subprotocol by themselves. This is due to the fact that both 1 and 3 receive their workflow from a location that is not inside of the subprotocol as illustrated in Figure 4-7.

4.4.3 Navigation

The canvas provided for laying out a protocol workflow is unbounded, with the editor providing a transparent pane over it with a metaphor similar to that used by Adobe Acrobat. Users can pan and zoom around the interface in any direction, but panning is blocked, emulating a limited canvas size, if



(a) 1 and 3 cannot alone compose a valid subprotocol



(b) For the same reasons, these three blocks are not a valid subprotocol

Figure 4-7: Examples of Invalid Subprotocol Layouts

that movement would result in the view pane being more than one screen size further away than the closest UI element in each direction. The canvas is in a sense extended each time an element is moved further out than other elements, allowing the pane to scroll further in the new direction. This behavior is intended to provide unlimited space, but to prevent authors from scrolling far enough away from the existing block elements that it is difficult to locate them again.

The pane over the canvas can also be zoomed in and out, making the blocks and connections smaller or larger. Zooming can be accessed either by the application menus, or by scrolling the mouse wheel up or down. This zooming is intended to allow better protocol visualization, and help assist scrolling for those workflows which consist of large numbers of questions.

In addition to the direct methods of navigating the workflow interface, authors can also navigate around the canvas by interacting with elements in the Element Explorer. When the author selects an element by double-clicking it, the view pane is panned such that any block corresponding to that element is brought to the center of the screen and selected. If there are multiple blocks which correspond to that element, they can be cycled through by continuing to double-click the element.

4.4.4 Visual Affordances

There are a number of visual affordances that were included in the design of the workflow editor in order to enhance the system's clarity and learnability.

- Block Connection Indicators

  The nodes at the end of a block's connectors have a few affordances to imply what connections between blocks are possible. Nodes at the head of a block are drawn in an easily distinguishable convex polygon shape, while nodes at the exits of a block are drawn similarly, but in a concave polygon shape. When these nodes are dragged together, they fit together like a puzzle piece, providing an affordance that they can be connected.

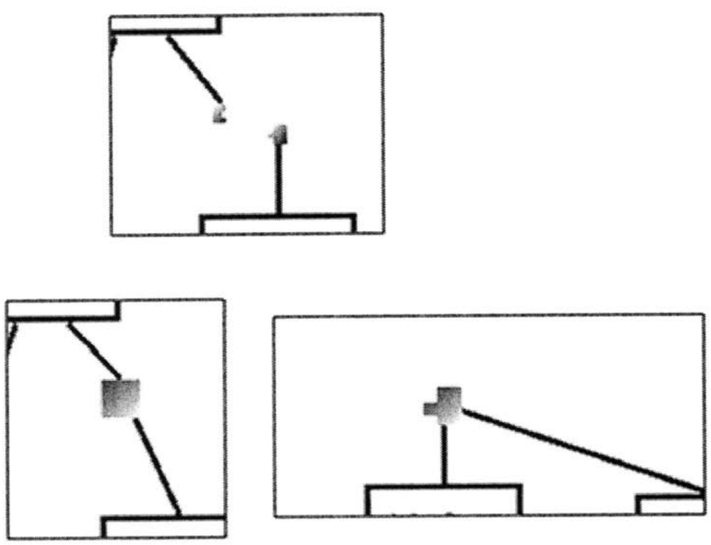


Figure 4-8: Block Connectors in three states. Top: Potential connection. Bottom-Left: Drop operation will connect. Bottom-Right: Invalid connection.

The system also provides feedback while performing the drag-and-drop operation that connects blocks together. When nodes are brought into proximity with each other, their two images are replaced by a single image which takes the shape of either a large green square, which implies a potentially succesful connection, or a large red version of the puzzle shape that is being dragged. This provides both a cue for when a drag operation becomes armed (when it can be completed by releasing the mouse), and a redundant cue for what connections are possible.

- Block Shapes

  Blocks in the system are drawn with different shapes in order to differentiate their types visibly. Some of these shapes are similar to those used by flowchart designers, but the overall design of the blocks is intended to be different enough from standard flowchart symbols that authors don't make the mistake of assuming that the workflow editor provides the same features as a flowchart

  The Root Block is drawn as a black rounded rectangle which contains the word "Start". This is similar to the symbol used at the start of a flowchart. Question Blocks are drawn as blue rectangles. A rectangle would represent a processing step in a flowchart, with a trapezoid being a closer analogue to the concept of a question. This divergence is intentional to differentiate the interface from a flowchart. Subprotocol Blocks are drawn as a large green square to represent the relatively larger size of their effect on the protocol's workflow. This shape has no special significance in flowchart notation.

4.4.5 Subprotocol Editing

In order to edit a subprotocol, the author can either manually select the edit option on the element explorer, or double-click an instance of the subprotocol on the workflow explorer. Editing a subprotocol takes place inside of a pane which lightboxes the workflow editor, and offers the same features and behaviors. The entry point to the subprotocol is defined by a root node, as is the case in the workflow editor, and the terminal edges of the subprotocol are defined by the unassigned exit paths from its blocks.

In addition to the editor pane for the subprotocol, buttons to either save or cancel the changes to the protocol are made visible. As was discussed while describing composing actions, not all configuraitons of blocks comprise a valid subprotocol. The cancel button is provided for authors to have a way to regain control of editing the main protocol workflow without saving changes to the subprotocol, which they may not be ready to do.

4.5 Debugging Interface

One important element of providing an end user programming environment is providing a test-bed in which programs can be run and tested for correct behavior. Scientia provides an interface for executing protocols on a mocked up screen, while guiding the author through the underlying representation to clarify questions of program state and help the author to identify and correct bugs.

When debugging begins, the application displays the workflow editor and selects the first block in the protocol, the one that is the child of the root block. Additionally, a window is popped up with displays a simulated device that can be used to interact with the protocol as if it were on a real device.

4.5.1 Simulated Device

The debugging environment presents the protocol created by the author as it might appear on a device with a small color screen like a cell phone or a PDA, both of which are popular targets for decision support tools. This screen is essentially visually identical to the interface used to create the questions. The author interacts with the screen as if it were a client application that they were a user of, and the widgets and controls presented on that screen are standard elements.

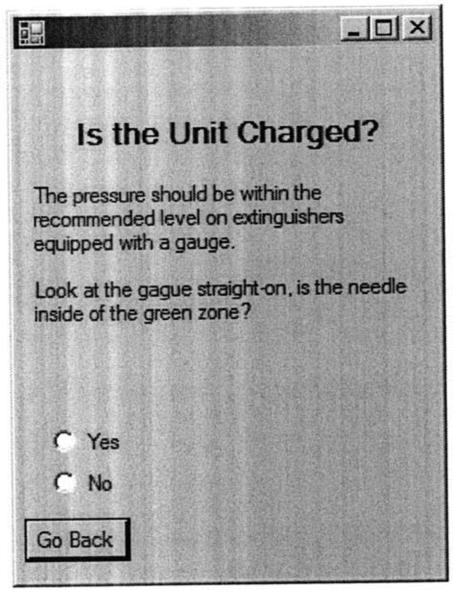


Figure 4-9: An example of a protocol running in the debugging interface

4.5.2 Navigation

While the debugging user answers questions, the workflow editor displays the current question as a selection, and follows the path through the user interface that corresponds to the users choices. The user can also seek backwards in order to identify what alternative paths would be available given different choices. When the user reaches an unassigned branch of a block's possible paths, the user is informed that the current path has ended with a prompt outside of the simulated device, to minimize confusion about the state of the protocol.

Chapter 5

System Architecture

Scientia is implemented in the Common Language Runtime managed environment, written in C#.

5.1 Client Architecture

The Scientia client is a standard desktop software application which is intended to prototype the interface for the creation, saving, and exporting of protocols. The design of the client code is divided into three main sections. The first section is a straightforward hierarchical collection of question data, the second is a digraph model for the representation of the workflow tree for protocols, and the third is a GUI decorator wrapper for handling user interface code.

The modularity and extensibility in the Scientia code architecture is primarily focussed on being able to extend the variety of responses that can be used by protocol authors, keeping in mind that one of the most broadly differing aspect of many decision support systems are what type of input is received for each step. The breadth of these responses is covered in more detail in the Question Types section of the Interface Design chapter.

The question and workflow representation seperation is provided both for ease of extensibility of question types as well as for reinforcing the types of protocols that are created in the system. Scientia was designed to help create protocols which are heavy on decision logic rather than data colleciton. As such, both the precise content of questions and the specific structure of the workflow tree are represented in a manner where both can be extended independently.

5.2 Identifier Generation

As described in the User Interface Design chapter, Scientia strives to remove the necessity for authors to create unique identifiers for questions due to their contribution towards producing user errors. In order to do this, the application creates a dynamic label for the question which is displayed in relevant locations like the explorer and the workflow designer.

The algorithm for generating a label for a given question is based on the content of the question's prompt, and has two steps. The first step, local trimming, is to attempt to distill the most relevant words from the prompt based only on the local information in the question and a static set of common words which are unlikely to add substantial context for the author to identify the question. The second step, neighborhood trimming, also takes into account the relative frequency of words in the heirarchical group in which questions reside and removes additional words which do not substantially distinguish between a question and its group neighbors.

The algorithm begins by composing and maintaining a table of occurrences and then frequencies of words that are used in authored questions. Identifier maintenance is triggered across all questions any time one is changed. Question objects are then responsible for constructing their own identifier based on the frequency tables it receives, one for the global set of words to be used and one for the local set. The question is then able to perform trimming against the static dictionary of frequencies and against the relative frequencies of its neighbors. Once a question has selected an ID, other questions are not able to take that id unless it comprises the same full text as that question. The algorithm is triggered by any changes to question text or structure of the question group heirarchy and acts globally across all identifiers.

This algorithm is not necessarily the optimal method of constructing question identifiers. Notably, it is possible that questions will not be able to locally construct the optimal id for itself in relation with the other questions in the protocol, since it is impossible for an individual question to know whether it is the best question to take a particular ID. This was not perceived as a serious problem, since the likelihood of identifier collision appeared to be small given the protocols examined during the problem analysis.

5.3 Protocol Representation

Scientia's architecture relies on its own internal model representation for questions and the protocol workflow without specifying or conforming to an external protocol representation. Although a number of different electronic protocol representation languages were identified and discussed in the Related Work chapter, it is beyond the scope of this work to select a particular representation for electronic protocols.

The protocols generated in Scientia are capable of producing representations in many possible languages, although the prototype does not have the ability to import any existing protocol representations without a loss of data. The issue of creating and exporting protocols which can be edited externally and then re-imported into the software without data loss is a substantial challenge and is left for further work.

Chapter 6

Evaluation

In order to evaluate the strategies used for addressing the difficulties that were identified in creating computer based protocols, we conducted a summative evaluation of Scientia's user interface. This evaluation took place in the form of user trials that were conducted with potential protocol authors and had the goal of identifying whether or not the subjects would be able to create a protocol that they had come up with, and whether or not the individual techniques used in the design of the user interface were effective at helping to mitigate some of the problems the protocol authors might encounter.

6.1 Trial Structure

6.1.1 Conditions

Volunteer subjects were chosen for the trials who had expert knowledge in a field that they worked in, but not necessarily in fields where they had created decision support protocols. Users were tasked with creating a full and tested decision support protocol of a length of their choosing that would represent their knowledge, and allow someone to reach a result state successfully that reflected the knowledge of the protocol's author.

The trials were conducted on a laptop computer that was running the Scientia software on Microsoft Windows and using a laptop keyboard and an external two-button mouse with a scroll wheel. The trials were conducted in the environment in which the subjects regularly worked. All of the subjects were familiar with using normal desktop software in Microsoft Windows.

Users were free to use any materials that they had on hand to help them codify their knowledge or as aids while doing so, which two of them took advantage of. Subjects were given a brief introduction as to the nature of decision support, and how it related to their field. They were also given an overview that the goal of the prototype was to help them to create a decision tree protocol that would allow someone to take advantage of their knowledge and informed of the nature of the trial that was being conducted.

6.1.2 User Classes

Trials were run with three subjects in distinct sessions. Each of the subjects was from a different potential class of protocol authors and exposure to decision support. The fields of research represented by the subjects in the study were Public Health and Museum Curation. The following classes of protocol authors were well represented by the subjects in these trials.

1. Low Protocol Exposure - Users who generally do not use decision support in their field, and do not have a strong familiarity with existing decision support techniques or technologies. Decision support isn't common in their field, and they do not necessarily have an expressed desire to utilize decision support systems. They may or may not be skilled computer users, but are generally not software developers.

2. Protocol Expertise - Users who are experts in a field and have written protocols before in a non-electronic format that were intended to be used by non-experts. These authors have a desire to use decision support in their field, and may be in a field where non-electronic decision support tools are common.

3. Electronic Protocol Experience - Users who have authored protocols before in electronic formats using some representation or using tools to help create those formats. These authors may have less experience with their field, or less broad experteise in it, but have more experience with attempting to codify protocols using software.

6.1.3 Trials

During the trials, subjects were asked to create a decision support protocol using Scientia's interface. Specifically, they were asked to take a process that they were familiar with and required their knowledge of their specific field and attempt to use the interface to create a decision tree that could be followed by a non-expert. The goal was for the outcome of the protocol tree to be of use to a non-expert at making a decision, or gaining information, that would be clear to an expert given the same information.

Before the trial had begun, the subjects were informed about the nature of Scientia's interface non-interactively and about the specific capabilities of the system. They were informed that during the course of the trial, their use of the software would be observed. Subjects were asked to narrate their experience using the software out loud while they were using it, including when they were having difficulties determining how to perform a task in the interface. They were told that although interactive help wouldn't be available to them during the trial, they would eventually be informed if a feature they were attempting to locate was not available in the software.

Once the trials had begun the subjects were asked to begin making a protocol, and to inform the observer when they had finished their protocol, felt that they could not effectively complete the protocol they were attempting to make, or wanted to end the trial for any other reason.

6.1.4 Interviews

After the trials were completed, the subjects were interviewed to discuss their experiences using Scientia's user interface. They were asked questions about their approach to designing their protocol in the system, and qualitative questions about some of the features in the system. In particular, subjects were asked to discuss their impressions of what the software was capable of and what they felt was missing.

6.1.5 Survey

As the final step of evaluation, subjects were asked to fill out a short survey about the trial. They were asked to quantitatively rank their experience with the software in both broad metrics like satisfaction and in more specific metrics like how well the system assisted them in handling the complex portions of their protocol.

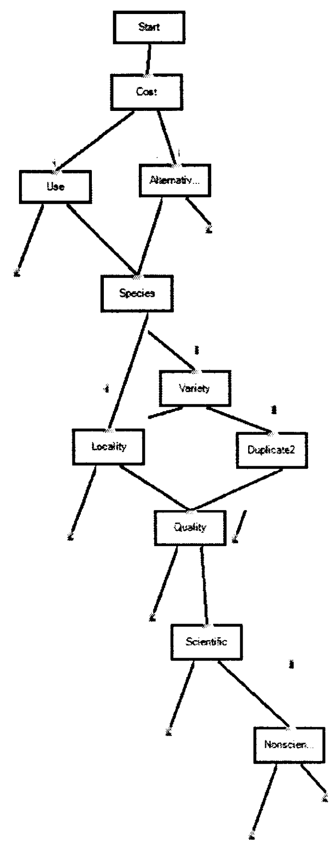


Figure 6-1: A protocol being created during one of the trials

The study included 10 questions. Six of the questions were feelings scales regarding how they felt about the interface's success in various dimensions, and 4 were free response questions which largely regarded to how capably they felt that the interface could be used.

All of the scales asked participants how they felt about a dimension of the interface on a 7 point scale, with 1 representing the worst, 4 representing a neutral feeling, and 7 representing the most positive. The respondents were told to rank their feelings using the interface against any common method for creating protocols that they might normally use, paper or electronic. If the subjects had little or no experience ever designing a protocol, they were informed that they could respond that any questions were not applicable to their experience.

6.2 Summative Results

Overall the results of the user trials conducted were mixed. Although the trials had a fairly high rate of success in being able to author protocols, the resulting protocols were not of particularly noteworthy complexity or size. Users seemed positive about the experience using the interface, especially in terms of ease, but were not confident that the interface would be able to let them create all of the types of protocols that they would create in the future.

6.2.1 Protocols

During the three user trials conducted, two of the three subjects were able to arrive at a protocol which satisfactorally represented the protocol which they intended to author when they began using the system.

None of the protocols were of a particularly large size, although all of them were greater than 10 questions overall. Only one of the protocols had more steps in the protocol than overall number of questions to be asked. There was a mixture of decision tree structures, with one protocol being largely linear with branching at the end, one protocol being strongly unbalanced, and one with multiple branches and an overall bushy structure.

The protocol which was unable to be satisfactorally completed required a lengthy boolean evaluation involving the answers to nearly every created question which would have grown the complexity of the tree to an unreasonable level. Although technically the protocol could have been represented as described with the feature set used, the interface would require an unreasonably large amount of questions to do so.

Survey Question	Mean Score out of 7
General Feeling about the Interface	5.5
Ease Of Use	5.33
Ease of Testing or Reading	5.5
Managing Many Questions	4.33
Ease of Handling Complexity	3
Accuracy and Specificity of Protocol	5.66

Table 6.1: Average Scores from Quantitative Evaluation

6.2.2 Direct Feedback

The results of the evaluative surveys filled out by trial participants provided a fairly consistent view of what portions of the interface were successful in representing the author's protocols and which were insufficiently featured. The full original surveys are provided in Appendix A. A summary of the mean score for the responses to each of the quantitative questions is provided in Table 6.1.

All study participants felt that the interface was able to their protocol specifically, and accurately, with a mean response score of 5.66. Similarly, all respondents felt that the interface was generally easy to use to create their protocol, despite receiving little to no instruction on how to do so. Some respondents did report some frustration with some aspects of the prototype interface not being implemented with the deep features of production software.

Scores for managing scale were mixed, but on average slightly positive. The score for the ease of testing and reading the protocols was high, and respondents expressed in their free responses strong positive feelings about the debugging and testing feature. This corroborates the observation during the trials that subjects were able to find bugs in their protocols and also provides support for the finding that respondents felt that their protocol was represented with high specificity due to being able to run the protocol and ensure that the appropriate paths were followed. A very common source of suggestions and feedback during interviews was related to additions and improvements to the debugging interface.

Despite two out of the three sessions resulting in a complete protocol, none of the respondents felt confident that the interface could be used out of the box to create the breadth of protocols that they would want to make, and two felt that could not. The respondents primarily cited a lack of support for individual features of logic that they would want or need to use. One respondent needed the ability to branch based on logic blocks which reviewed previous questions. Another needed to be able to ask multiple-selection questions, and also to branch based on ranges in integer-masked free-text. Extending the interface to be able to handle more arbitrary sets of logic could serve as a valuable contribution of future work.

6.3 Feature Evaluation

The success of some of the features of the interface were evaluated after the trials using the notes collected while observing the trials, the survey responses, and the interviews with subjects.

6.3.1 Workspace Layout

The direct manipulation interface was generally successful in clearly representing the branching logic for the subject's protocols. All of the authors were able to discover the model for laying out a protocol by directing the outcome of individual questions. Additionally, the authors reported fairly positive levels of satisfaction with using the interface.

The workflow interface caused difficulty for each author in multiple ways, however. Many of the affordances were successful in prompting the appropriate action to be taken, like putting together the ends of each transition by fitting together nodes and breaking edges by hovering over them and clicking the razor blade. However, a common source of confusion was attempting to manipulate the interface in intuitive, but unsupported, ways which were not caught during unstructured formative evaluations, like dragging blocks into a position where the pieces fit together in the expectation of actively linking them.

The flexibility of the workflow interface met with mixed success. Two of the three subjects laid out their protocol manually in a custom way and reported that they appreciated doing so. Users also reported that they wanted, and somewhat expected, the interface to provide more structure in the form of a grid or other specific format, though, or used the interface in such a way that implied that the spatial orientation of the blocks would influence the structure of the protocol.

Many of the workspace's features were not discoverable without prompting. The marquee selection method for multiple blocks, for instance, was not discovered by any of the subjects. One subjects attempted to use a keyboard+mouse drag to marquee select blocks, but did not discover the correct key to use. Two subjects reported wanting to use a marquee selection while authoring their protocol. All of the subjects discovered and utilized panning, and one subject discovered and used the zoom in/out feature.

6.3.2 Question Creation

Of all of the questions created across trials, single selection with two answers was overwhelmingly the most common question type created, although not all of those two responses fit into the same category of "Yes" and "No". Free text was used along with a numeric mask.

Although question grouping in the element explorer was not a feature that was utilized, it was a feature that two of the subjects reported expecting to use for long-term use of the application.

6.3.3 Automatic Identifiers

Users did not have trouble distinguishing their questions using distinct automatically generated identifiers. This result was conflated, however, to some degree by the ways in which they were used during the trials. The text being used for the identifiers did not include the fully expected question text due to text being entered elsewhere, so many of the identifiers were generated in a way that was already easy to distinguish. In some cases, for example, the title of questions were short two word fragments like

"Is Coughing?", rather than more full text sentences like "Is the Patient Coughing?", with the full text for the question being filled out more completely.

All of the subjects reported that they did not want to write their own question identifiers. None reported that they expected to be able to provide question identifiers as seperate values. No subjects reported difficulties associated with the dynamic identifiers changing in response to wording changes or expressed a desire for the id to stay static after being generated.

6.3.4 Debugging Interface

All of the subjects utilized the debugging interface to help debug the workflow logic they had created while authoring their protocol. One subject reported that the concept of executing a protocol in a test bed wasn't immediately intuitive. All of the subjects identified errors in their protocol while using the debugging interface. The most common problems identified were that the wrong branch was taken while debugging or that the text of a given question or its response was incorrect.

There were mixed reactions to the format of the debugging interface, but the most common criticism was that it was difficult to see what the outcome of choices would be before they were made. Exploring the most effective set of debugging features for decision support protocols is a rich field of potential future work.

Chapter 7

Conclusions

In this thesis, I designed, implemented, and evaluated a user interface to assist experts in creating decision support protocols which could be used electronically.

The interface design followed user centered design processes. Problem analysis was performed by analyzing existing decision support protocols and establishing the difficulties inherent to creating an unambiguous electronic version of the decision support logic in them.

The interface was implemented as a desktop application running on top of the Common Language Runtime managed software environment.

The interface was tested with a summative evaluation in the form of user trials with experts from different user classes. The overall outcome was that using the interface was a positive experience, and that the interface was partially succesful, and could represent two out of three of the subject's sample protocols. It is clear from evaluation that the set of possible questions or blocks that can be authored will need to be expanded to include concepts like blocks which can perform logical evaluations on existing responses in order to be more useful for broadly representing complex protocols.

7.1 Contributions

Many of the approaches to managing scale, complexity, and specificity from the interface design were successful in enabling protocol authors to create the protocols they had designed in a way that could be used electronically.

Scientia's interface was designed with a focus on creating protocols as a decision support interaction, and attempted to abstract away most of the specifics of representing the protocol as structured data or as a software program. Most of the successes of the interface were realized in the features tailored for this focus: like creating a protocol as a set of questions and outcomes, abstracting away unique identifiers, and a visual style based on the outcome of the question and workflow design, rather than its abstract representation. I hope that the success of this approach will influence the design of other end user programming environments.

Users reported high satisfaction with the direct manipulation interface, and the prototype interface was able to comfortably represent the protocol intended for the two out of three subjects. Although not all of its features were discoverable without instruction, the use of a visual canvas or chart to represent the flow of decision support logic was intuitive and straightforward to authors regardless of their level of exposure to decision support protocols. Although Scientia is not capable of representing highly complicated protocols, the results of this evaluation imply that there exist a number of experts who could successfully create electronic protocols pertinent to their fields with such an interface without needing to learn a specification language or develop their protocol in an abstract format.

7.2 Future Work

Many of the features implemented in Scientia were not evaluated strongly during user trials and warrant further investigation.

The use of automatically generated dynamic identifiers was largely successful in abstracting away the notion of a unique id and of providing a visually distinct reference to identify questions, but none of the user trials dealt with protocols at the upper end of scale in the low-hundreds of questions. Similarly, there exist many possible methods for generating such ids based on word frequency and uniqueness. The idea of using these ids to manage a very large number of user generated items effectively would serve as a valuable contribution for end user programming systems that currently require identifiers or names for aspects of the system.

The use of modularity in the form of composed sub protocols was another feature that was not strongly represented in the evaluation of the interface. During post-trial interviews, however, subjects commonly reported that for some protocols that they could imagine that they would like to be able to re-use portions of their decision logic in multiple areas. One user expressed an interest in another prospective feature of the composed sub protocols, which was to reuse them across multiple protocols. The realization of this idea is akin to the concepts of library linking in software development, and share many of the same issues, such as synchronising changes between multiple instances of when a composed chunk is used. Determining the utility of this idea in a long-term protocol design environment would provide valuable insight as to whether the concept of modularity extends comfortably to the mental model of protocol authors.

The debugging interface provided by Scientia was of high value for identifying problems with protocols, but was not the focus of extensive user centered design. All of the authors provided suggestions for potential debugging interface features. Most authors wanted to be able to either jump into debugging at specific parts of a protocol, and at least one author wanted to be able to provide answers to large numbers of questions at once and see the resulting state. Determining the most valuable ways to help authors ensure the integrity of their decision support protocols is a good avenue for increasing the utility of electronic authoring systems.

Appendix A

Raw Evaluation Responses

This appendix contains the text of the survey provided to trial subjects for evaluation of the user interface after completing each user trial, along with each subject's responses.

1) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the interface?

5

2) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your normal method of designing forms in terms of ease of creating a protocol?

6

3) In what ways was it better? Saves time by not having to start with writing html codes, the drag and drop is nice and easier to understand, Make easy to rearrange the order of questions which happens a lot when interacting with field users, It gives the possibility of running the test in the same window without opening the emulator in a separate file, Saves time to not write and even remember how to write all the open and closing tags which always make someone spend a lot of time debugging just in case something is missing.

4) In what ways was it worse?

No save as option which forces someone to finish creating the form when start, Can lose your work very easily if anything happen such as power problem. Help menu doesn't work which make it difficult to learn more about the interface, the mapping method of using the nodes is challenging to figure out, doesn't include multiselect question options which is needed many times.

5) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your old method of designing forms in terms of testing or reading a protocol?

5

6) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about creating and managing large numbers of questions with the interface?

4

7) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the handling complex aspects of your protocol with the interface?

3

8) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about how specifically and correctly your protocol was represented by the interface?

6

9) Do you think that you could design the forms you need to use with the interface?

No

10) If not, what aspects of form design are missing from the overall interface?

Figure A-1: Subject #1 Survey Responses

The aspect of the forms that are missing on this interface are multiselect options, binding which include relevancy and constraints conditions. On my xform that I created using the interface, I did not need the calculations so I am not sure if it is possible to do that using the interface. So I suggest based on my experience of making xform, it is important the interface include constraints, and calculations. For example if someone want to limit a certain number of integer, let say I want the answer in a certain field be not less than 10. Or If I want the question Y be displayed only if the answer of question X is greater than 15. Also think if you can include calculations in the interface to make it even more useful because right now, I am sure if there are codes that handles calculations very well on xform.

Subject #1 Survey Responses

1) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the interface?

5.5

2) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your normal method of designing forms in terms of ease of creating a protocol?

6

(though i don't actually make protocols, but i'll pretend i crafted them by hand in xforms..and if i needed to do something that this system couldn't do, then i'd give is a 1 or 2.)

3) In what ways was it better?

it was simple and quick to do the straightforward things. i liked the visualization of the branching logic. having the ability to run the questions is very great.

4) In what ways was it worse?

mostly in terms of being limited in terms of what you could do, e.g., not being able to do more logic.

5) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your old method of designing forms in terms of testing or reading a protocol?

6

6) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about creating and managing large numbers of questions with the interface?

5

7) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the handling complex aspects of your protocol with the interface?

2

8) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about how specifically and correctly your protocol was represented by the interface?

6

9) Do you think that you could design the forms you need to use with the interface?

not without substantial additions

10) If not, what aspects of form design are missing from the overall interface?

ways of managing the complicated logic.

Figure A-2: Subject #2 Survey Responses

1) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the interface?

6. It seemed easy to use and I think it has real potential.

2) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your normal method of designing forms in terms of ease of creating a protocol?

4. I don't normally design forms.

3) In what ways was it better?

It's being structured seems an advantage.

4) In what ways was it worse?

Not worse. The lack of even simple instructions was surprizing.

5) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how does the interface compare to your old method of designing forms in terms of testing or reading a protocol?

NA

6) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about creating and managing large numbers of questions with the interface?

4 I'd need more experience to answers usefully.

7) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about the handling complex aspects of your protocol with the interface?

4 I don't think I got to a level anyone would term complex!

8) On a scale from 1-7 with 1 being the worst, 7 being the most positive, and 4 being neutral how did you feel about how specifically and correctly your protocol was represented by the interface?

5 I think the protocol could be well represented but it would take time to refine.

9) Do you think that you could design the forms you need to use with the interface?

5 Yes, if I needed to design a form. Some practice would be helpful.

10) If not, what aspects of form design are missing from the overall interface?

The lack of any instructions is an initial problem.

Figure A-3: Subject #3 Survey Responses

Appendix B

Selected Protocols

The Child/Adult Diarrhea Decision Trees were authored and graciously provided for analysis and publishing by Lindsay and Dan Palazuelos.

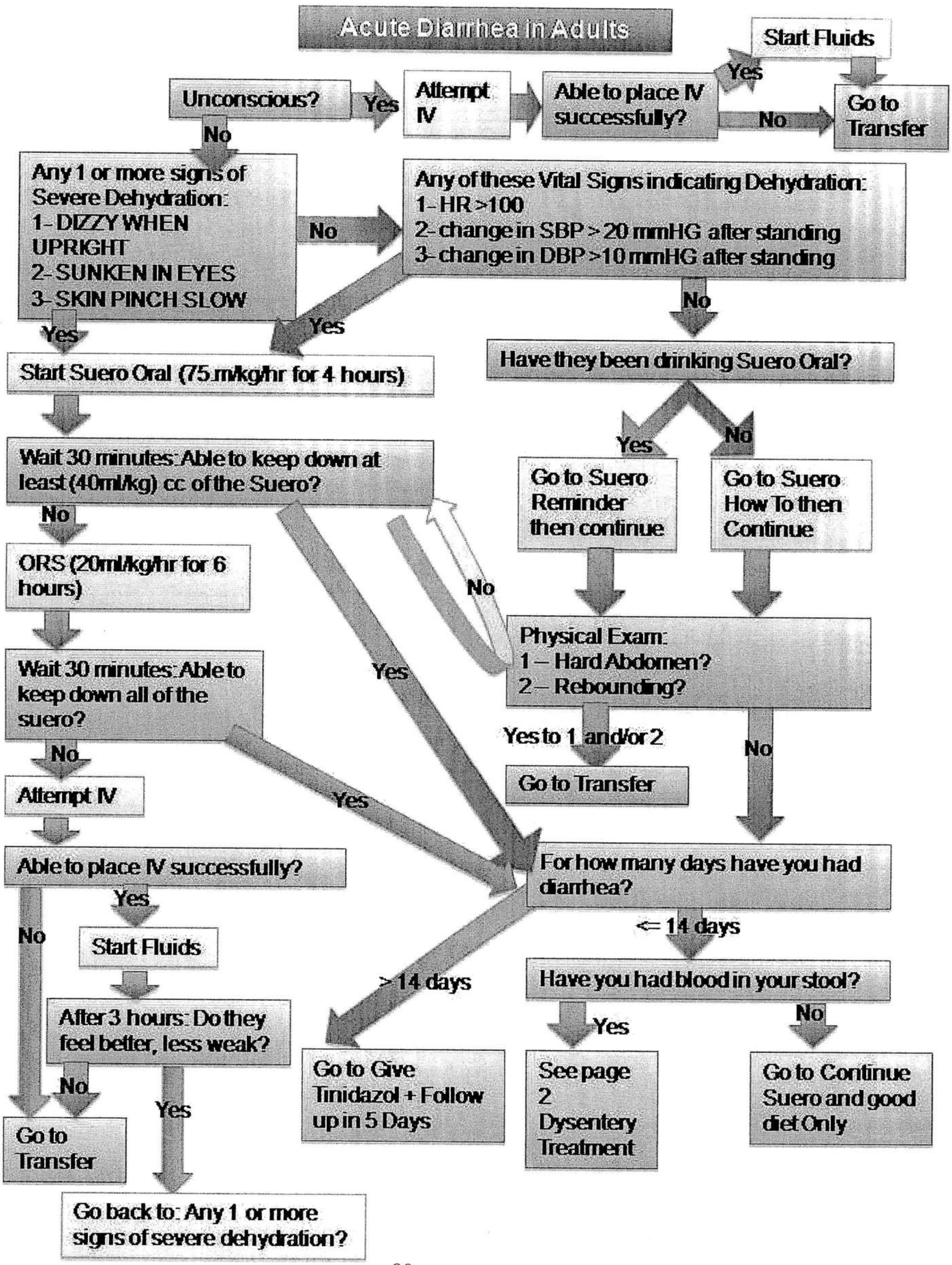
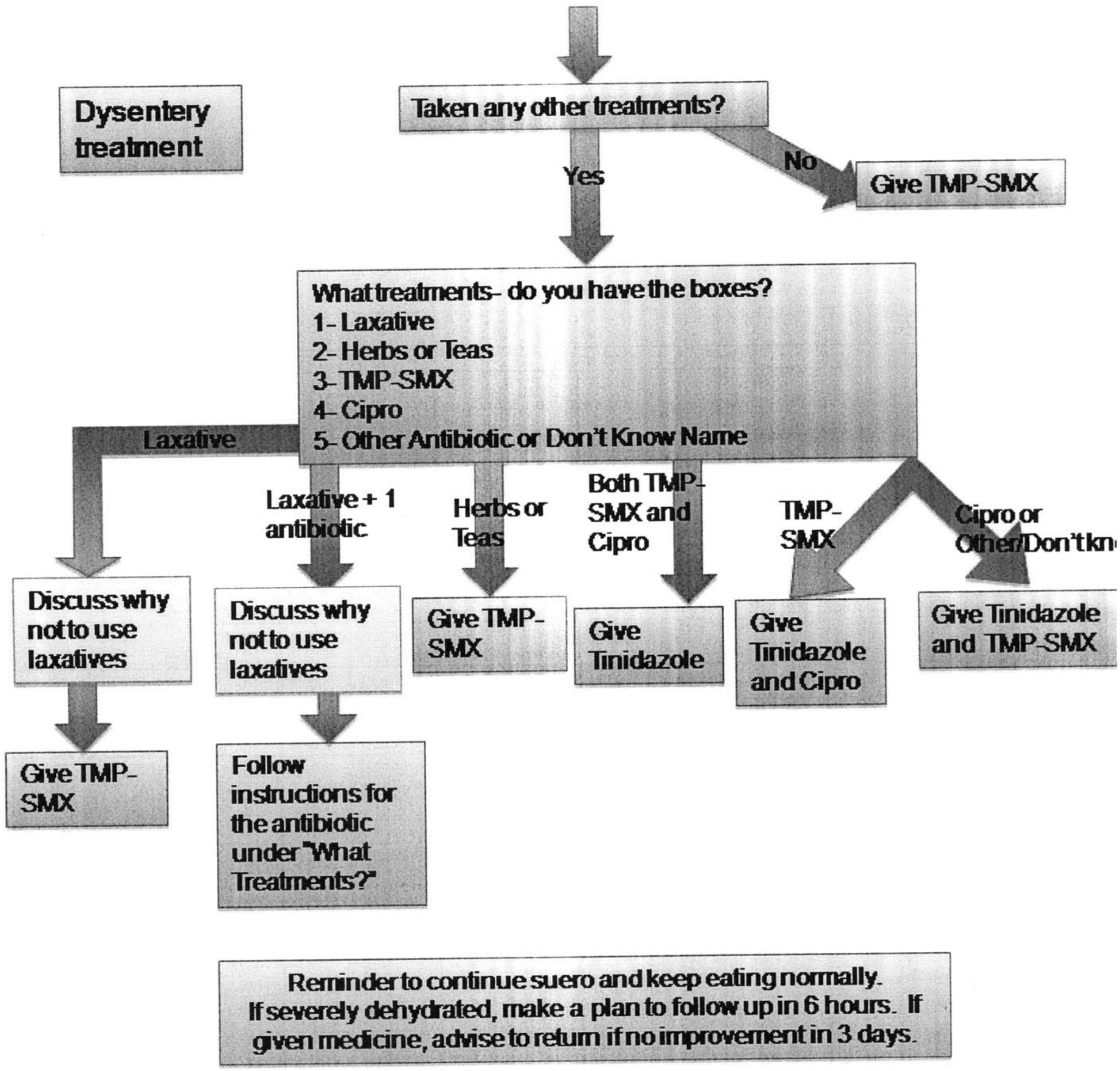


Figure B-1: Flowchart Format Adult Diarrhea Decision Protocol



Flowchart Format Adult Diarrhea Decision Protocol

Item	Item Type	Item Format	Content	Options	Action
A	Question to CHP	Y/N	Is the person unconcious?	Yes	Go to B
				No	Go to F
B	Instruction	Basic	Attempt to place an IV. Remember to use alcohol to clean the skin.	Continue	Go to C
C	Question to CHP	Y/N	Were you able to place the IV successfully?	Yes	Go to D
				No	Go to E1
D	Instruction	with Calculation	Start fluids at ??????	Continue	Go to E1
E1	Final Instruction	Basic	This person is severely dehydrated needs to be transferred	Stop	
E2	Final Instruction	Basic	This person may have a problem with their bowel and needs to be transferred	Stop	
F	Question to CHP	Check Box	Does this person show any of the following signs of severe dehydration?: a) dizzy when upright b) sunken in eyes c) skin pinch goes back slowly	Yes to Any	Go to G
				No to All	Go to BB
G	Instruction	with Calculation	Start Suero Oral at .75ml/kg/hr for 4 hours	Continue	Go to H1
H1	Instruction	Basic	Ask the person to lie down for a physical exam.	Continue	Go to I1
H2	Instruction	Basic	Ask the person to lie down for a physical exam.	Continue	Go to I2
I1	Question to CHP	Check Box	Does this person have any of these symptoms? A) Hard Abdomen B) Rebounding?	Yes to Any	Go to E2
				No to All	Go to J
I2	Question to CHP	Check Box	Does this person have any of these symptoms? A) Hard Abdomen B) Rebounding?	Yes to Any	Go to E2
				No to All	Go to R

Figure B-2: Spreadsheet Format Adult Diarrhea Decision Protocol

J	Question to CHP	Y/N	Is the person able to keep down most of the Suero Oral?	Yes	Go to R
				No	Go to K
K	Instruction	with Calculation	Slow down the Suero Oral to 20ml/kg/hr for 6 hours	Continue	Go to L
L	Instruction	Basic	Ask the person to continue drinking for 30 minutes. Wait with them.	Continue	Go to M
M	Question to CHP	Y/N	After 30 minutes is the person able to keep down all the suero?	Yes	Go to R
				No	Go to N
N	Instruction	Basic	Attempt to place an IV. Remember to use alcohol to clean the skin.	Continue	Go to O
O	Question to CHP	Y/N	Were you able to place the IV successfully?	Yes	Go to P
				No	Go to E1
P	Instruction	with Calculation	Start fluids at ?????? Monitor the person for 3 hours.	Continue	Go to Q
Q	Question to Patient	Y/N	After 3 hours, do you feel somewhat better, less weak?	Yes	Go to F
				No	Go to E1
R	Question to Patient	Type in Number	For how many days have you had diarrhea?	>14	Go to S
				</= 14	Go to T
S	Final Instruction	with Calculation	This person likely has amebiosis. Give them Tinidazol and make a plan in your calendar to see them again in 5 days.	Stop	
T	Question to Patient	Y/N	Have you had any blood in your stool?	Yes	Go to V
				No	Go to U

Spreadsheet Format Adult Diarrhea Decision Protocol

U	Final Instruction	Basic	Antibiotics and other medicine won't be effective for this type of diarrhea. Advise the person to continue making and drinking suero oral. They should drink a cup after each time they go to the bathroom. They should also keep eating a little bit every few hours, even if they don't feel like it.	Stop	
V	Question to Patient	Y/N	Have you taken any other treatments?	Yes	Go to X
				No	Go to W
W	Final Instruction	with Calculation	Give this person TMP-SMX to cure their dysentery.	Continue	Go to ???
X	Question to Patient	Check Box with Combinations	What treatments- can you show me the boxes? Choose all that apply: a) laxatives b) herbs or teas c) TMP-SMX d) Cipro e) other antibiotic f) don't remember	a	Go to Y1
				a + c, d, e, or f	Go to Y2
				b	Go to W
				c	Go to S
				c and d	Go to Z
				d, e or f	Go to AA
Y1	Instruction	Basic	Advise against laxatives. Diarrhea dehydrates the body, and laxatives actually make even more water leave. They will usually make things worse and can even be dangerous.	Continue	Go to W
Y2	Instruction	Basic	Advise against laxatives. Diarrhea dehydrates the body, and laxatives actually make even more water leave. They will usually make things worse and can even be dangerous.	Follow antibiotic instructions	

Spreadsheet Format Adult Diarrhea Decision Protocol

Z	Final Instruction	with Calculation	Give this person both Tinidazol and Cipro.		
AA	Final Instruction	with Calculation	Give this person both Tinidazol and TMP-SMX.		
BB	Question to CHP	Check Box	Does the person have any of these vital signs?: 1) Heart Rate >100 2) change in SBP >20mmHG after standing 3) change in DBP >10mmHG after standing	Yes	Go to G
				No	Go to CC
CC	Question to Patient	Y/N	Have you been drinking suero oral?	Yes	Go to DD
				No	Go to EE
DD	Instruction	Basic	Encourage the person and remind them to continue drinking suero.	Continue	Go to H2
EE	Instruction	Basic	Tell the person how to make suero oral: in one liter of clean water, stir in one handful of sugar, one pinch of salt, one pinch of bicarbonato and the juice of half a lemon or orange	Continue	Go to H2

Spreadsheet Format Adult Diarrhea Decision Protocol

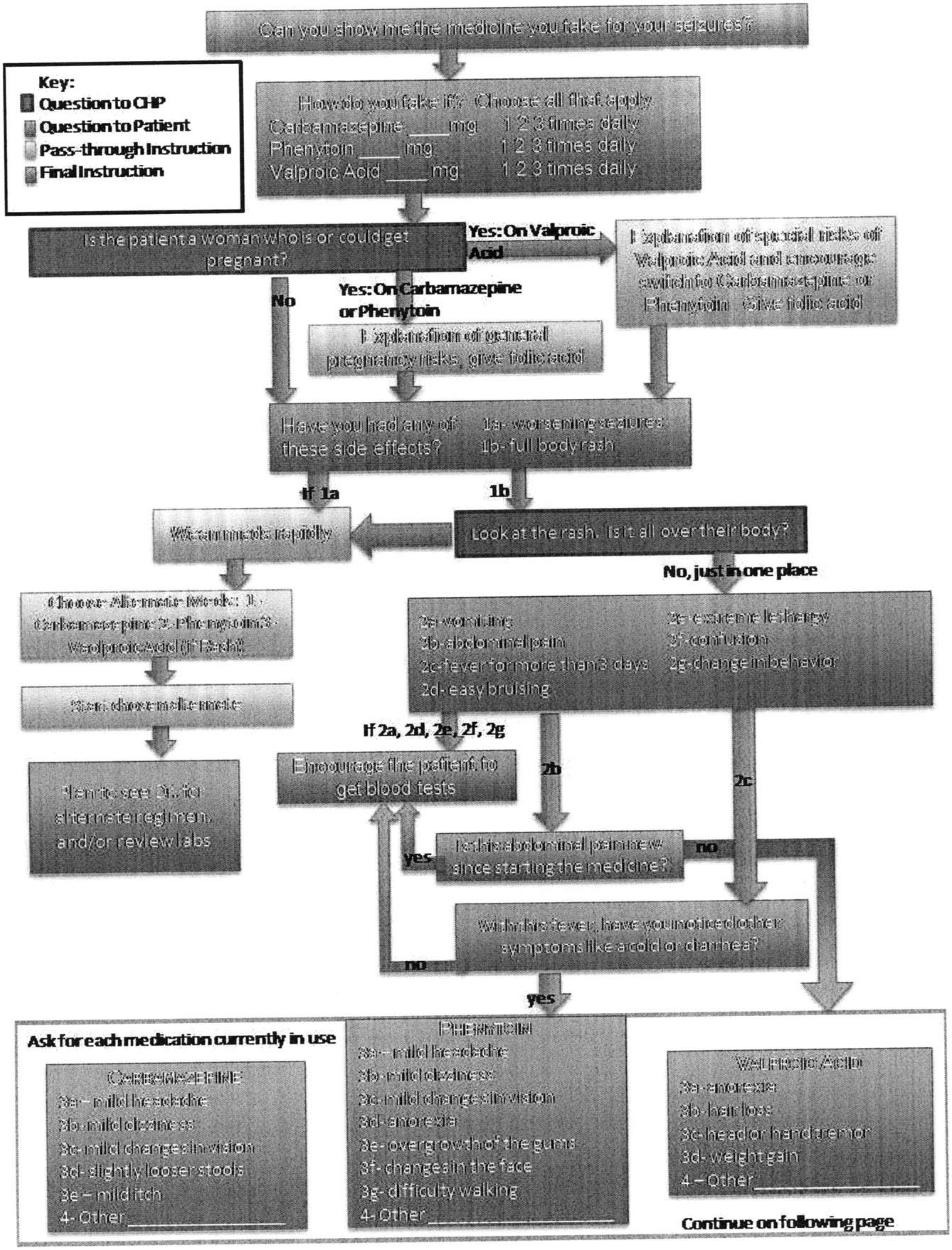
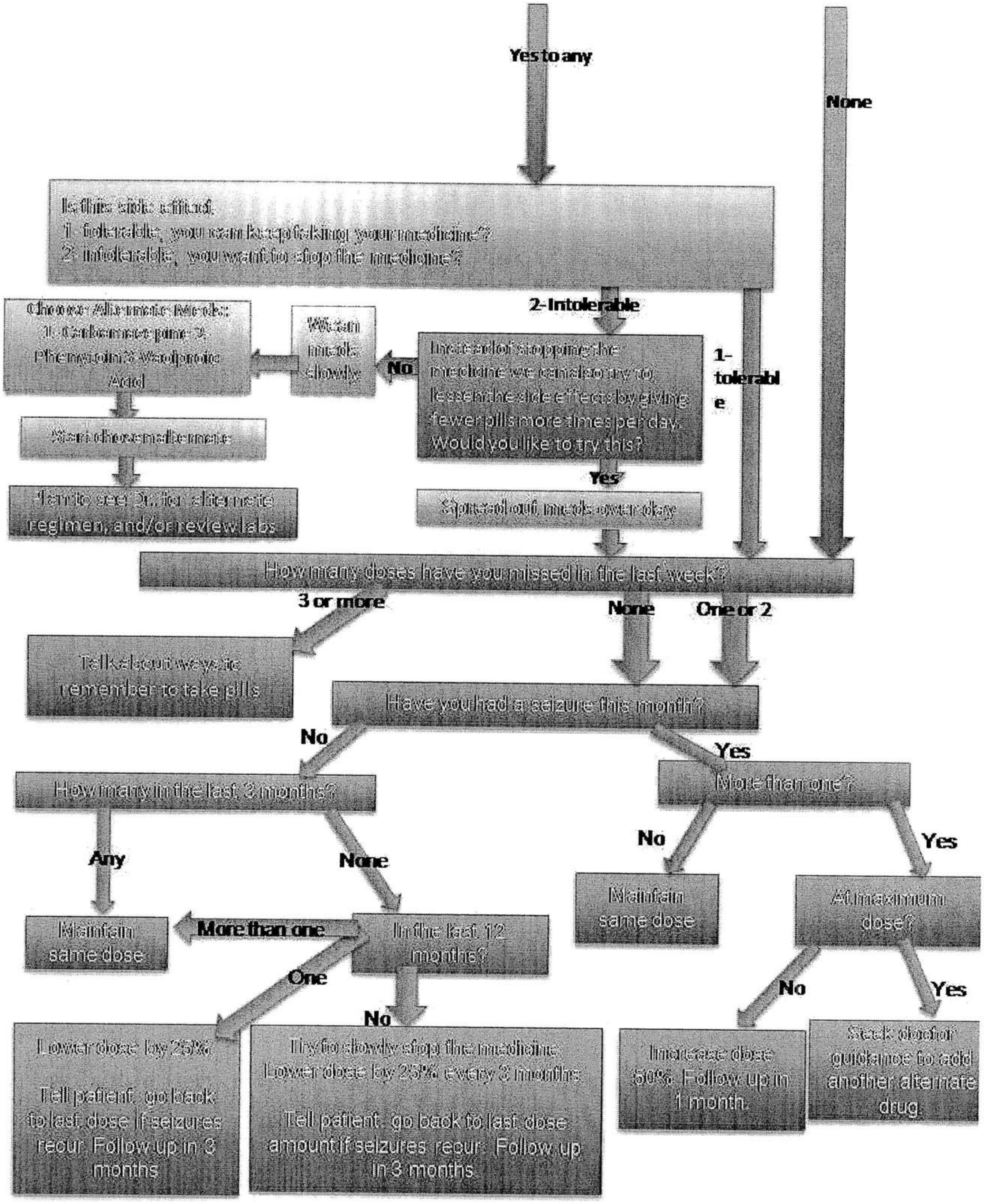


Figure B-3: Flowchart Format Adult Epilepsy Decision Protocol



Flowchart Format Adult Epilepsy Decision Protocol

Item	Item Type	Item Format	Content	Options	Action
A	Instruction to Patient	Basic	Can you show me the medicines you take for your seizures?	Continue	Go to B
B	Question to CHP/ Patient	Check box	Which of these do you take? 1) Carbamazepine 2) Phenytoin 3)	Continue	Go to corresponding C1-3
C1	Question to Patient	Type in numbers	How do you take it? Carbamazepine ___mg ___ times daily	Continue	Save and Go to D after C1-3 complete
C2	Question to Patient	Type in numbers	Phenytoin ___mg ___ times daily	Continue	Save and Go to D after C1-3 complete
C3	Question to Patient	Type in numbers	Valproic Acid ___mg ___ times daily	Continue	Save and Go to D after C1-3 complete
D	Question to CHP	Y/N	Is the patient a woman who could get pregnant?	Carbamaezpine or Phenytoin	Go to E
				Yes + Valproic Acid	Go to F
				No	Go to G
E	Instruction	Basic	Explain pregnancy risks of both seizures and seizure medication. Give folic acid and advise on how to take.	Continue	Go to G
F	Instruction	Basic	Explain special pregnancy risks for Valproic Acid. Encourage to switch to a new medication. Give folic acide and advise on how to take.	Continue	Go to G
G	Question to Patient	Check box	Have you been having either of these side effects? 1a- worsening seziures 1b- rash	1a	Go to H
				1b	Go to L
				None	Go to M

Figure B-4: Spreadsheet Format Adult Epilepsy Decision Protocol

H	Instruction	With Calculation	Begin Rapid Weaning of Medication	Continue	Go to I
I	Question to CHP/Patient	Check Box	Choose Alternate Medication 1- Carbamazepine 2- Phenytoin 3- Valproic Acid, if Rash	Continue	Go to J
J	Instruction	Basic	Start Chosen Alternate	Continue	Go to K
K	Final Instruction	Basic	Plan to see physician for alternate regimen	Stop	
L	Instruction	Y/N	Look at the rash. Is it all over their body?	Yes	Go to H
				No, only in one place	Go to M
M	Question to Patient	Check Box	Have you been experiencing any of these side effects? 2a-vomiting 2b-abdominal pain 2c-fever for more than 3 days 2d- easy bruising 2e-extreme lethargy 2f-confusion 2g-change in behavior	2a, c, d, e, f, g	Go to N
				2b only	Go to O
				2c only	Go to P
				None	Go to corresponding Q
N	Final Instruction	Basic	Encourage the patient to get blood tests	Stop	
O	Question to Patient	Y/N	Is this abdominal pain new since starting the medicine?	Yes	Go to N
				No	Go to corresponding Q
P	Question to Patient	Y/N	With this fever have you noticed any other symptoms like a cold or diarrhea?	Yes	Go to corresponding Q
				No	Go to N

Spreadsheet Format Adult Epilepsy Decision Protocol

Q1	Question to Patient	Check Box	Carbamazepine. Have you been experiencing the following as a side effect? 3a – mild headache 3b- mild dizziness 3c- mild changes in vision 3d- slightly looser stools 3e – mild itch 4- Other ______________	Yes to Any	Continue through corresponding Q2-3 and/or then go to R
				None	Continue through corresonding Q2-3 and/or then Go to V
Q2	Question to Patient	Check Box	Phenytoin. Have you been experiencing the following as a side effect? 3a – mild headache 3b- mild dizziness 3c- mild changes in vision 3d- anorexia 3e- overgrowth of the gums 3f- changes in the face 3g- difficulty walking 4- Other ______________	Yes to Any	Continue through corresponding Q3 and/or then go to R
				None	Continue through coresonding Q3 and/or Go to V
Q3	Question to Patient	Check Box	Valproic Acid. Have you been experiencing the following as a side effect? 3a- anorexia 3b- hair loss 3c- head or hand tremor 3d- weight gain 4 – Other______________	Yes to Any	Go to R
				None	Go to V

Spreadsheet Format Adult Epilepsy Decision Protocol

R	Question to Patient	Check Box	Is this side effect 1- tolerable, you can keep taking your medicine? 2- intolerable, you want to stop the medicine?	1	Go to S1
				2	Go to S2
S1	Question to Patient	Y/N	We can also try to lessen the side effects by having you take fewer pills more times per day. Would you like to try this or keep things the way they are?	Yes	Go to U
				No	Go to V
S2	Question to Patient	Y/N	Instead of stopping the medicine we can also lessen the side effects by having you take fewer pills more times per day. Would you like to try this?	Yes	Go to U
				No	Go to T
T	Instruction	With Calculations	Begin Slow Weaning of Medication	Continue	Go to I
U	Instruction	Basic	Spread same dosage of medication out over the day	Continue	Go to V
V	Question to Patient	Type in numbers	How many doses would you say you've missed in the last week?	None	Go to X
				1 or 2	Go to X
				3 or more	Go to W
W	Final Instruction	Basic	Discuss ways to remember to take pills.	Stop	
X	Question to Patient	Y/N	Have you had a seizure in the past month?	Yes	Go to Y
				No	Go to DD
Y	Question to Patient	Y/N	More than one?	Yes	Go to Z
				No	Go to CC

Spreadsheet Format Adult Epilepsy Decision Protocol

Z	Question to CHP	Y/N	Is this patient at the maximum dose of their medicine?	Yes	Go to AA
				No	Go to BB
AA	Final Instruction	Basic	Seek Dr. guidance to add another drug.	Stop	
BB	Final Instruction	With Calculations	Increase daily dose by 50%. Follow up in one month.	Stop	
CC	Final Instruction	Basic	Maintain the same dose and follow up in three months.	Stop	
DD	Question to Patient	Type in Number	How many in the last three months? ____	None	Go to EE
				Any	Go to CC
EE	Question to Patient	Type in Number	How many seizures in the last year?	None	FF
				1	GG
				>1	CC
FF	Final Instruction	With Calculations	Try to slowly stop the medicine. Lower the daily dose by 25% and plan to lower it another 25% every three months. Tell patient: go back to last dose level if seizures increase. Follow up in three months.	Stop	
GG	Final Instruction	With Calculations	Lower daily dose by 25% to begin to very slowly try to wean off the medicine. Tell patient: go back to last dose level if seizures increase. Follow up in three months.	Stop	

Spreadsheet Format Adult Epilepsy Decision Protocol

Bibliography

[1] A. A. Boxwala, M. Peleg, S. Tu, O. Ogunyemi, Q. T. Zeng, D. Wang, V. L. Patel, R. A. Greenes, and E. H. Shortliffe. GLIF3: a representation format for sharable computer-interpretable clinical practice guidelines. *J Biomed Inform*, 37:147–161, Jun 2004.

[2] M. A. Crawford, T. M. Harrington, T. V. Russell, F. A. Franklin, C. D. Lorish, T. J. Turner, and B. K. Baumann. Practice extenders and PDA-based counseling for smoking and unhealthy diet. *Ann Fam Med*, 3 Suppl 2:S41–43, 2005.

[3] Brian DeRenzi, Neal Lesh, Tapan Parikh, Clayton Sims, Werner Maokla, Mwajuma Chemba, Yuna Hamisi, David S hellenberg, Marc Mitchell, and Gaetano Borriello. E-imci: improving pediatric health care in low-income countries. In *CHI '08: Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, pages 753–762, New York, NY, USA, 2008. ACM.

[4] John H. Gennari, Mark A. Musen, Ray W. Fergerson, William E. Grosso, Monica Crubzy, Henrik Eriksson, Natalya F. Noy, and Samson W. Tu. The evolution of protg: An environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58:89–123, 2002.

[5] Jesper Kjeldskov and Jan Stage. Designing the handheld maritime communicator. In *DUX '03: Proceedings of the 2003 conference on Designing for user experiences*, pages 1–15, New York, NY, USA, 2003. ACM.

[6] Andre W. Kushniruk, Marc M. Triola, Elizabeth M. Borycki, Ben Stein, and Joseph L. Kannry. *International journal of medical informatics*, volume 74, chapter Technology induced error and usability: The relationship between usability problems and prescription errors when using a handheld application, pages 519–526. Elsevier Science Ireland Ltd.,, Aug 2005.

[7] Aswath Manoharan, Jeannie Stamberger, YuanYuan Yu, and Andreas Paepcke. Optimizations for the ecopod field identification tool. Technical Report 2007-22, Stanford InfoLab, March 2007.

[8] M. Mitchell, N. Lesh, M. Cranmer, H. Fraser, I. Haivas, and K. Wolf. Improving care - improving access: the use of electronic decision support with aids patients in south africa. *International Journal of Healthcare Technology and Management*, 10(3):156–168, 2009.

[9] Lucila Ohno-machado, John H. Gennari, Shawn Murphy, Nilesh L. Jain, D. Sc, Samson W. Tu, Diane E. Oliver, Edward Pattison-gordon, Robert A. Greenes, Edward H. Shortliffe, Ph. D, Ph. D, Ph. D, Ph. D, Ph. D, and G. Octo Barnett. The guideline interchange format: A model for representing guidelines. *Journal of the American Medical Informatics Association*, 5:357–372, 1998.

[10] Lindsay Palazuelos and Dan Palazuelos. *Adult Diarrhea Treatment Protocol.* Partners in Health. Protocol is provided in multiple formats in Appendix B.

[11] Lindsay Palazuelos and Dan Palazuelos. *Children Diarrhea Treatment Protocol.* Partners in Health. Protocol is provided in multiple formats in Appendix B.

[12] Lindsay Palazuelos and Dan Palazuelos. *Epilepsy Management Protocol.* Partners in Health. Protocol is provided in multiple formats in Appendix B.

[13] Brendan M. Reilly, Arthur T. Evans, Jeffrey J. Schaider, Krishna Das, James E. Calvin, Lea Anne Moran, Rebecca R. Roberts, and Enrique Martinez. Impact of a Clinical Decision Rule on Hospital Triage of Patients With Suspected Acute Cardiac Ischemia in the Emergency Department. *JAMA*, 288(3):342–350, 2002.

[14] Edward H. Shortliffe. A rule-based computer program for advising physicians regarding antimicrobial therapy selection. In *ACM '74: Proceedings of the 1974 annual ACM conference*, pages 739–739, New York, NY, USA, 1974. ACM.

[15] World Health Organization. *Integrated Management of Adult Illness (IMAI).*

[16] World Health Organization. *Integrated Management of Childhood Illness (IMCI).*

[17] World Health Organization. *Pregnancy, Childbirth, Postpartum and Newborn Care: A guide for essential practice.*