The formalization of medical protocols:

easier said than done.

"According to my new computerized diagnostic software, you need to upgrade your kidneys, defragment your liver, and make a back-up copy of your spleen."

The formalization of medical protocols:

easier said than done.

Amsterdam, 16 December 2002

Marije Geldof
mgeldof@cs.vu.nl

Department of Artificial Intelligence
Vrije Universiteit Amsterdam
De Boelelaan 1081a
1081 HV Amsterdam

Supervisors
Frank van Harmelen
Annette ten Teije

Abstract

The role of medical protocols is becoming more and more important in the medical field. Within the Protocure project it has been shown that the quality of medical protocols can be improved by formalization, something much easier said than done. The formalization turns out to be a very time-consuming task resulting in a formal protocol that is much more complex than the original version and the relation with this original protocol remains unclear. This graduation project was intended to improve the insight in formal protocols and possibilities to the speed up of the formalization process.
The Guideline Markup Tool, developed to support the formalization process, has been used to define and visualize all the relationships between a formal protocol and its original version. During this linking task the tool has also been evaluated for its usability.
The most surprising result of the linking is that it uncovered anomalies that had been introduced during the formalization process. Furthermore the linking provided a division between the parts of the formal protocol that are related to the original protocol and the parts that aren't. This is an important indicator for the causes of the increased complexity of formal protocols.
Finally reappearing patterns, whose reuse could support the formalization process in the future, have been identified in the formal protocols.
Using the Guideline Markup Tool and its linking functionality, resolving the complexity issues and using the identified patterns all turn out to be challenges for improving the formalization process.

Acknowledgements

"*Alone we can do so little; together we can do so much*" – Helen Keller

I also couldn't have done this graduation project on my own:
not without the guidance and help of my supervisors Annette ten Teije and Frank van Harmelen,
not without the answers of Mar Marcos to all my questions about the protocols,
not without the pleasant communication and co-operation with Peter Votruba about GMT,
not without all the other members of the Protocure project,
not without the love and personal support of Robert and my parents,
and finally not without all the other people I should have mentioned.

Thanks!

Contents

1 Introduction

"Doctors are men who prescribe medicines of which they know little, to cure diseases of which they know less, in human beings of whom they know nothing." - Voltaire

Clinical guidelines and protocols are gaining increasing acceptance in medical practice as a means to support both diagnosis and treatment. They are becoming increasingly important in health-care because of their potential to promote high-quality practice and reduce variations in care while improving cost efficiency [1]. However in order to reach these goals, protocols need to be of high quality, something easier said than done. Anomalies like ambiguity and incompleteness appear frequently in medical guidelines [2]. Besides, guidelines can be inconsistent because of the lack of familiarity of the designer with certain principles and notations. The medical community has tried to address this problem of guideline quality improvement, but their efforts are not sufficient because they mainly rely on informal processes and notations [3]. An appropriate representation language with a clear and well-defined semantics allows for a more systematic verification of guidelines by formal methods.
The goal of the Protocure project [4] is to improve medical protocols by formal methods. Several languages for representing medical protocols exist [5], but within the Protocure project the formal representation language Asbru [6] is used. Two medical guidelines have been selected and successfully modeled in Asbru. The first is the American Academy of Pediatrics practice guideline for the Management of Hyperbilirubinemia in the Healthy Term Newborn [7]. The second is the Dutch College of General Practitioners (NHG) standard for Type 2 Diabetes Mellitus [8].
Translating clinical protocols to formal guideline representations is not an easy task however [9]. The text guidelines must be adhered to, while at the same time adding information from common knowledge of physicians. The translation is done by knowledge engineers and they could use tools that support the task.
One of the tools that is being developed is a tool that will not only make it possible to add elements to an Asbru guideline but also show the user which parts of the original guideline the information comes from. This makes it possible for the medical practitioner to understand decisions made by the execution unit in terms of the original guideline. This Guideline Markup Tool (GMT) is an editor that helps translating guidelines from text into Asbru [10].
The first aim of this graduation project is to use the Guideline Markup Tool to define the relations between parts of the original protocol and the Asbru version to get a better insight in how the two are related. An indirect result of this task is a second aim of this project namely the evaluation of the Guideline Markup Tool with respect to the linking task.
In practice it turns out that the Asbru translations of a protocol are much more complex than the original protocol. It is not completely clear yet what causes this complexity; there are only some hypotheses. Hopefully the relationships between the original protocol and its Asbru translation will give more insight into this matter. The third aim of this project therefore is to determine the origin of this complexity.
Translating protocols into Asbru is quite a time consuming task. That is why there exists a growing interest in ways to speed up this process. One way to do this is to define design patterns. A design pattern can be seen as a standard model for standard modeling situations. The standard model only needs to be slightly adapted to make it suitable for a specific situation.
By defining these design patterns as macros in for example GMT, they can be easily reused in the practice of modeling medical protocols. The last aim of this project is to identify these patterns in Asbru protocols.

Summarizing the goal of this project is four-part:

- defining the relations between the original protocol (jaundice and diabetes) and its Asbru translation using the Guideline Markup Tool (GMT)
- evaluation of the Guideline Markup Tool with respect to the linking task
- determining where the increased complexity of the Asbru protocols (compared to the original protocol) comes from
- identifying patterns in Asbru protocols

This thesis is structured as follows:
Chapter 2 provides the background of this project. First of all it explains what medical protocols are. Then the Protocure project, the selected reference protocols, the representation language Asbru and the formalization process carried out are discussed. Finally the Guideline Markup Tool that has been used in this graduation project is described.
Next the different goals described above will be discussed separately.
Chapter 3 discusses the linking of protocols with the Guideline Markup Tool. On one hand it discusses the linking itself and on the other hand the observations obtained by the linking.
Chapter 4 evaluates the Guideline Markup Tool considering the linking task. It describes what has already been improved, what should be improved and what possibly could be improved.
Chapter 5 deals with all the different aspects responsible for the explosion in size of Asbru protocols compared to their original protocol.
Chapter 6 presents the patterns that have been identified in the protocols that were selected for the Protocure project.
Finally **chapter 7** summarizes the work done and **chapter 8** gives some recommendations for future work.

2 Background

Medical protocols often presume its users to have certain background knowledge, not having to explain everything in complete detail. The next chapters, describing the work done during this graduation project, also presume certain background knowledge. This chapter is meant to provide the reader with this background knowledge. So readers that are already familiar with the Protocure project can only have a quick look at this chapter and then continue to chapter 3.
First of all the reader will have to know what medical protocols are, which is explained in paragraph 2.1. Next it is important to know what the Protocure project stands for, what can be read in paragraph 2.2. Within this project two reference protocols have been selected, discussed in paragraph 2.3, the representation language Asbru is used, described in paragraph 2.4 and the formalization process is carried out, which is discussed in paragraph 2.5. Finally paragraph 2.6 deals with the Guideline Markup Tool, a tool that has been utilized in this graduation project.

2.1 What are medical protocols?

"*Medicine is a science of uncertainty and an art of probability.*" Sir William Osler

The quote above gives a very good characterization of the medical field. In contrast with mathematical and natural science, medicine is a scientific field, which embodies unsolved and unforeseeable problems, i.e. the domain knowledge does not cover the aspects with full certainty. For example if a certain medicine is administered to a patient it is never completely sure in advance what the effect will be. Of course it is known what the effect will most likely be, but there is always a certain chance of unexpected circumstances, for example hypersensitivity to the administered medicine.
Some characteristics of the medical domain are [11]:

- Large domain knowledge is available, but this knowledge is often uncertain and vague
- Incomplete and non-deterministic information about the world state and effects of actions
- Unobservable, underlying processes influence the observable state of the world (e.g. events occur without actions preceding them)
- Actions and effects are not necessarily instantaneous: actions are often continuous (have duration) and might have delayed effects.
- Parallel and periodic execution of plans is common and plans can also be suspended
- A goal may not be achievable in time

Presently the medical world is facing two major problems. First of all an increasing amount of information for making clinical decisions resulting from modern equipment becomes available and paradoxically this makes the decision making more difficult [12]. More is known about the human body, how it works, and how it fails, but having too much, poorly organized information, can cause as many errors in decisions as having too little information. Second of all there exists a need to improve the quality of health care through increased awareness of proper disease management techniques [11].
Clinical guidelines or protocols can help to overcome these problems. Clinical practice guidelines can be defined as: systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances [13]. They can be viewed as reusable skeletal plans that need to be refined when applied to a particular patient [14]. Skeletal plans are plan schemata at various levels of detail that capture the essence of a procedure, but leave room for execution time flexibility in the achievement of particular goals [15]. However guidelines will not address all the uncertainties of current clinical practice and should be seen as only one strategy that can help improve the quality of care the patients receive [16].

Simply said clinical guidelines describe how a medical practitioner should act under certain circumstances for certain patients. For example when a patient suffers from complaint A, the guideline advises the practitioner to administer drug B. Some equivalent terms used for clinical guidelines in the medical field are clinical practice guidelines, medical guidelines and medical protocols. Although these terms sometimes carry different meanings, they will be used indistinctly in this document.
Medical guidelines can be developed for a wide range of subjects in the medical field. The methods of guideline development should ensure that treating patients according to the guidelines will result in the outcomes that are desired. The development requires sufficient resources in terms of people with a wide range of skills and a systematic review of the evidence available at that moment. The resulting guidelines are usually written in text combined with some additional formats like tables, flowcharts and graphs.
The application of the guidelines requires a medical practitioner to collect and interpret considerable amounts of data, to apply standard therapeutic or diagnostic plans and to revise the plans when necessary. Furthermore the guidelines often involve implicit assumptions about the knowledge of the practitioner executing the plan. So the practitioner shouldn't just indiscriminately execute the guideline, but also use his common sense.
Clinical guidelines have potential benefits as well as harms [17]. Because of the potential of their benefits, guidelines are accepted more and more in the medical field. The most important benefits of guidelines are:

- Promotion of high quality practice
- Reduction of variations in care
- Improvement of cost efficiency

However these benefits only apply if the guidelines satisfy strong quality requirements [18]. Inside the medical world already a lot of attempts have been made to guarantee this quality. Since most of these attempts use informal methods and notations, they turn out to be inadequate. The Protocure project, which will be discussed in the next paragraph, uses a different approach, namely the quality improvement of medical protocols by formalization.

2.2 The Protocure project

"Medicine is a collection of uncertain prescriptions the results of which, taken collectively, are more fatal than useful to mankind." - Napoleon Bonaparte

Protocure is a European project carried out by five different institutions that focuses on "*improving medical protocols by formal methods*" [4]. It is a multidisciplinary project since every participating institute is specialized in another aspect of the project, amongst others the medical, the formalization and the verification aspect.
Presently medical protocols are described by a combination of different formats like text, flow diagrams, tables, etc. This makes them easy to read for medical practitioners without training in formal languages, but also has a number of disadvantages, the most important being that they are often incomplete and ambiguous and lack any formal means of verification [19]. The idea of using formal methods is to uncover these ambiguous, incomplete or even inconsistent parts of the protocols, by defining all the different descriptions more precisely using a formal language and to enable verification.
This requires the definition of an adequate representation language, the development of techniques for the formal analysis of protocols translated in this representation language and the evaluation of the feasibility of the approach based on the formalization and verification of real-life medical protocols.
Considering the first two aspects some of the participating institutions have already been working on the Asbru language for protocol description [20] and on the KIV interactive verification system [21]. The third aspect, the evaluation of the feasibility of the use of formal methods for quality improvement of protocols, is the main objective of this project. The steps that are carried out for this evaluation are as follows:

1. Two reference protocols that cover a wide variety of protocol characteristics are selected.
2. The two protocols are formalized using the representation language Asbru.
3. Formal semantics for Asbru elements are defined in KIV.
4. Desirable/required protocol properties are identified with the help of medical experts.
5. The Asbru formalizations are translated into KIV.
6. The selected protocol properties are verified for the KIV translations of the protocols.
7. Evaluation of the verification results, e.g. the amount of errors (expected or unexpected) that can be uncovered in this way are determined.

The work that has been done for this graduation project is mostly related to the first two steps. That is why the following paragraphs will only deal with aspects related to those steps. The next paragraph will discuss the two selected reference protocols. Subsequently paragraph 2.4 will deal with the representation language Asbru. And finally paragraph 2.5 will describe the process of translating a medical protocol in Asbru in more detail.

2.3 Selected reference protocols

"They do certainly give very strange, and newfangled, names to diseases." - Plato

The Protocure project considers two protocols: a general practitioners protocol for the management of diabetes mellitus type 2 [8] and a pediatrics protocol for the management of jaundice in the healthy term newborn [7]. They have been developed by the Dutch Association of General Practitioners (NHG) and the American Academy of Pediatrics (AAP) respectively.

2.3.1 Diabetes mellitus type 2

Diabetes mellitus is a chronic disease that requires long-term medical attention both to limit the development of its devastating complications and to manage them when they do occur. It can cause serious health complications including heart disease, blindness, kidney failure, and lower-extremity amputations.
Three forms of the disease can be distinguished: type 1, type 2 and gestational diabetes. Diabetes mellitus type 2 is the most common form accounting for 90 to 95 percent of all people diagnosed with diabetes mellitus. Normally insulin signals the body's tissues to remove sugar from the blood. However in case of diabetes mellitus type 2 the cells of the body do not use insulin effectively, which results in too much sugar in the bloodstream, leading to symptoms like: fatigue, frequent urination and unexplained weight loss. There are two possible causes for this abnormal high blood sugar level:

- not enough insulin is being made
- the body's cells are not responding properly to insulin.

Once diabetes mellitus type 2 is diagnosed the goal of treatment is to control the level of sugar in the blood. The sugar level should stay in the same range as for a non-diabetic person. This is amongst others done by:

- measuring blood sugar regularly
- meal planning
- exercise
- medicine

All the protocols developed by the Dutch Association of General Practitioners (NHG) are available in two versions: a detailed version and a so-called “summary-card”, a short overview of the detailed version. The Protocure project mainly uses this summary-card and the detailed version is only used as a reference.
The summary-card is 3 pages long and contains knowledge mainly in the form of text, enhanced with two tables. One table for the diagnosis of diabetes mellitus type 2 based on glucose and the other for the monitoring of glucose once a patient is diagnosed with diabetes mellitus type 2. Furthermore the diabetes protocol refers to a table with the risk of getting coronary heart diseases, which has been also used in the translation to Asbru.
The protocol consists of a diagnosis (or evaluation) part and a policy (or treatment) part, to be performed in sequence. As soon as a patient is diagnosed to have diabetes mellitus type 2, the treatment of the patient starts. In some cases the patient is referred to another specialist like a chiropodist. The idea behind this is that the protocol is only covering the management of diabetes mellitus type 2 and all other tasks are delegated.
An important part of the protocol is the table used to monitor the glucose values. It is used to determine the adequate treatment for a diabetes patient.

2.3.2 Jaundice (hyperbilirubinemia)

Jaundice (or hyperbilirubinemia) is a common disease in newborn babies. Under certain circumstances, too high bilirubin levels may have harmful neurological effects. In many cases jaundice disappears without treatment, but sometimes phototherapy is necessary to reduce the levels of total serum bilirubin (TSB), which indicates the presence and severity of jaundice. Sometimes, jaundice can also be a sign of another severe disease.
The protocol is 10 pages long and contains knowledge in various forms, namely:

- Text (this is the main body of the protocol)
- A list of factors to be considered when assessing a jaundiced infant (for instance, ethnicity with a significant chance of hemolytic disease)
- Two tables, one for the management of jaundice in the healthy term newborn and the other for the treatment options for breast-fed infants
- A flowchart representing the steps described in the guideline.

The protocol consists of an evaluation (or diagnosis) part and a treatment part, to be performed in sequence. During the application of the protocol, as soon as the possibility of a more serious disease is uncovered, the recommendation is to abort without any further action. The idea behind this is that the protocol is exclusively intended for the management of jaundice in healthy term newborns.
An important part of the protocol is the table used to determine the adequate treatment from the TSB value and the age of an infant.

2.3.3 Protocol dimensions

There exists a wide variety of protocols and they can differ on several dimensions related to among other things the content or form of the protocol. Important dimensions are [22]:

- The target users of the protocol (e.g. general practitioners or nurses)
- Aspects of clinical practice it covers (e.g. diagnosis and/or treatment)
- The period of time which the protocol applies to (varying between short time-span (hours to days) and long time-span (weeks to years).
- The form; mostly more or less structured textual descriptions, sometimes combined with additional formats, such as tables, lists or flowcharts.

When the reference protocols for this project were selected, their variance was an important selection criterion. The two selected protocols not only come from different organizations, but also differ on the described dimensions. The table below compares the two protocols on these dimensions:

Jaundice	**Diabetes**
Management of jaundice in healthy term newborn	Management of diabetes mellitus type 2
Diagnosis and treatment	Diagnosis and treatment
Short time-span application	Long time-span application
Text, lists, tables, flowcharts	Text, tables in 2 versions
Non-specialist/hospital use	General practitioner/consultation use

2.4 Asbru – a guideline representation language

"The art of medicine consists in amusing the patient while nature cures the disease." - Voltaire

The Asgaard (Asbru) project [23] is a project carried out by several international institutions that focuses on *"designing task-specific problem-solving methods to support the design and execution of time-oriented skeletal plans"*. It tries to bridge the gap between planning approaches and medical approaches, having regard for the demands of the medical staff on the one hand and applying rich plan management on the other hand. The project concentrates on supporting therapeutic issues.
Asbru is a plan representation language that is used in the Asgaard project to represent clinical guidelines as time-oriented, skeletal plans [24]. It can be used to express clinical protocols as skeletal plans that can be instantiated for every patient. Skeletal plans are "plan schemata at various level of detail, that capture the essence of the procedure, but leave room for execution-time flexibility in the achievement of particular goals" [15]. This means they are usually reusable in different contexts.
In Asbru a clinical guideline is represented as a set of hierarchical plans. A plan consists of a name, a set of arguments, including a time annotation (representing the temporal scope of the plan) and five elementary components: preferences, intentions, conditions, effects and a plan body, which describes the actions to be executed. The plan name is compulsory and all the other components are optional. Each plan may contain an arbitrary number of subplans within its plan body, which may themselves be decomposed into sub-subplans. So a plan can include several potentially decomposable concurrent or cyclical plans.
During execution time, the system interpreter attempts to decompose each plan into subplans until a non-decomposable plan – called an action or user-performed plan- is found. Such a plan is referred to the agent for execution, which may result in an interaction with a user or an external call of a program.
Within the Protocure project Asbru is considered as a semi-formal language to support the tasks necessary for protocol-based care. It is called a semi-formal language because its semantics, although more precise than in other protocol representation languages, are not defined in a formal way [22]. This semi-formal quality makes Asbru suitable for an initial analysis but not for systematic verification of protocols.
Apart from the language itself, meanwhile several tools have been developed (or are still under construction) to facilitate working with Asbru. For example a visualization and user interface for writing plans in Asbru [25], a user interface that allows users to enter specific data about a treatment [26] and an interpreter that is able to simulate the application of clinical guidelines written in Asbru [27]. At the moment the Guideline Markup Tool [10], which is also used in this graduation project, is being developed and will be discussed in more detail in paragraph 2.6.
Below the five elementary components and time annotation will be discussed separately [6]. Those aspects appearing in the Protocure protocols will be discussed in more detail and the aspects that do not appear will be mainly left out of consideration.

In the remaining chapters of this thesis examples from the Protocure protocols will be used to elucidate certain topics. These examples will not be presented in the for Asbru usual XML-format, but in an intermediate format that is much more readable.

2.4.1 Preferences

Preferences bias or constrain the selection of a plan to achieve a given goal and express a kind of behavior of the plan. When a choice has to be made between multiple subplans during the execution, preferences influence this selection. The following preferences can be distinguished:

1. `strategy`: a general strategy for dealing with the problem (e.g. aggressive, normal).
2. `utility`: a set of utility measures (e.g. maximize patient convenience).
3. `select-method`: a matching heuristic for the applicability of the whole plan (e.g. the extent to which the conditions of a plan have to be met).
4. `resources`: a specification of prohibited, discouraged or obligatory resources (e.g. patients can be hypersensitive to certain medicine. In this case they do more harm than good).
5. `responsible actor`: a set of actors, who are entitled to adapt the protocol (e.g. general practitioners, nurses).

Preferences don't appear in the two protocols used in the Protocure project.

2.4.2 Intentions

Intentions are high-level goals at various levels of the plan, represented as temporal patterns of provider actions and patient states, that should be maintained, achieved or avoided during execution. There are states and actions that should be true during execution or either once the execution has finished. This is not only important for selecting the right plan, but also for critiquing treatment plans as part of the ever-ongoing process of improving the treatment. This makes intentions one of the key parts of Asbru.
For example during the treatment of a diabetes patient the diabetes guideline prescribes the injection of insulin intended to lower the level of blood glucose. However the medical practitioner recommends the patient to reduce the intake of carbohydrate during dinner (which reduces the level of blood glucose). This seems to contradict the prescribed action of the guideline. Nevertheless the underlying intentions of both actions are the same, namely to reduce the level of blood glucose. So the medical practitioner is still following the intention of the protocol.
Four categories of intentions can be identified:

1. `intermediate state`: the patient state(s) that should be maintained achieved or avoided during the applicability of the plan (e.g. weight gain levels are slightly low to slightly high).
2. `intermediate action`: the provider action(s) that should take place during the execution of the plan (e.g. monitor blood glucose once a day).
3. `overall state`: the overall pattern of patient states that should hold after finishing the plan (e.g. patient has less than one high glucose value per week).
4. `overall action pattern`: the overall pattern of provider actions that should hold after finishing the plan (e.g. patient had visited dietitian regularly for at least three months).

In contrast to preferences intentions do appear quite often in the two protocols used in the Protocure project. An example of an intention appearing in the jaundice-plan ***Policy***, which is intended to avoid bad glucose values:

intentions
avoid-bad-glucose-values: **avoid intermediate-state**:
(glucose-monitoring = bad)

2.4.3 Conditions

Conditions are temporal patterns that need to hold at particular steps to induce a particular state transition of the plan instance. They are represented as logical expressions combined with logical operators. Asbru provides seven different conditions:

1. `filter-preconditions`: need to hold initially if the plan is applicable but can not be achieved (e.g. patient is suffering from diabetes). If the filter precondition is fulfilled, the plan becomes *possible* otherwise the plan is *rejected*.
2. `setup-conditions`: must hold before a plan can be executed. It can be achieved by actions from physicians (e.g. a cholesterol test). If the setup condition is fulfilled the plan becomes *ready* otherwise if it is not fulfilled within a predefined period the plan is *rejected*.
3. `activate-mode`: determines if the plan should be started manually or automatically. If the activate mode is `automatic`, the plan is started (e.g. it becomes *activated*) immediately after becoming *ready*. If it is `manual` the user is asked for approval before the plan is started. If the user does not answer in time the plan is *rejected*.
4. `suspend-conditions`: determine when an activated plan has to be suspended (e.g. blood glucose has been high for four days). If the suspend condition is fulfilled, the plan is *suspended*. Should the suspend condition become false later, the plan still stays suspended. Only if the reactivate condition is fulfilled, the plan can be reactivated
5. `reactivate conditions`: determine when a suspended plan has to be reactivated (e.g. blood glucose level is back to normal or is only slightly high). If the reactivate condition is fulfilled, the plan is *activated* again. If another plan was started because this plan was suspended, this plan is not activated as long as the other plan is in plan state *activated*.
6. `complete-conditions`: determine when a started or restarted plan is completed (e.g. the delivery has been performed). If the complete condition is fulfilled and all mandatory subplans of a plan have completed themselves the plan is *completed* (i.e.. it finishes successfully). As long as the activate condition is not fulfilled the plan stays active (if it is not aborted or suspended).
7. `abort-conditions`: determine when an activated or suspended plan has to be *aborted* (e.g. the dose of a medicine reaches a maximum value). If the abort condition is fulfilled, the plan is *aborted*, i.e. it fails. If this plan is a mandatory subplan of the plan, which activated it, that plan will also fail.

As well as intentions, conditions appear frequently in the two protocols used in the Protocure project. For example the diabetes-plan ***Policy***, that should only be executed for patients suffering from diabetes mellitus type 2, therefore has the following condition:

conditions
filter-precondition: (glucose-evaluation = DMT2)

2.4.4 Effects

Effects describe the expected behavior of the plan's execution, independent of the complete-condition and the intentions. They describe the functional relationship between the plan arguments and measurable parameters or the overall effect of a plan on parameters. It is possible to include a likelihood annotation to specify the probability of occurrence of these effects.
Just as for preferences effects have been included in order to make the selection of plans easier. Whenever there is a choice between different plans, the decision, which plan to choose can be based on the effects a plan will have.

Like for the preferences, effects don't appear in the two protocols of the Protocure project.

2.4.5 Plan body (actions)

The plan body is a set of plans to be executed in parallel, in sequence, in any order, or in a certain frequency. It contains the actions and/or subplans that will be executed as parts of the plan.
A plan-body can be of the following forms:

1. *subplans*: a set of plan steps performed in parallel or sequentially.
2. *cyclical plan*: a plan repeated several times.
3. *single-step*: a single step of plan execution, consisting either of a plan activation, a variable assignment or the setting of the context.
4. *user-performed*: a tag showing that this plan is executed through some action by the user, which is not further modeled in the system.
5. *to-be-defined*: a special tag declaring that this plan is not executable.
6. *refer-to*: a reference to the plan body of another plan.

The first four types of plan-body do appear in the two protocols in the Protocure project and will be discussed in more detail below.

2.4.5.1 Cyclical plan

The cyclical plan is a complex element, which implements various types of repetitive action. It consists of the following parts:

1. *start-time*: evaluates to a time point, at which the action specified by cyclical-plan-body is performed for the first time.
2. *cyclical plan body*: is a single plan step mostly a plan-activation. This step is repeated as specified in the repeat specification.
3. *any-repeat-specification*: specifies the times at which the step defined by cyclical-plan-body is repeated.
4. *set-of-cyclical-complete-conditions*: is a combination of cyclical complete conditions connected by Boolean operators.
5. *max-attempts*: is the number of attempts made to complete the action given in cyclical-plan-body before the plan aborts.

An example of a cyclical plan appearing in the diabetes-plan ***Treatments-and-Controls***, which represents the quarterly visits to a medical practitioner, is the following:

cyclical-plan Quarterly-control **retry-delay**: min = 10w, max = 14w

2.4.5.2 Subplans

The element subplans is used to group plan steps in one of the following temporal orderings:

1. *sequentially*: the steps (or subplans) are performed in strict order. Each plan is started after its predecessor is finished.
2. *parallel*: all steps or subplans are started at the same time. They may finish any time.
3. *any-order*: the steps are performed one at a time, without strict ordering. For example the second step can be performed before the first one. Yet at each instant in time, only one step is performed, i.e. their execution may not overlap.
4. *unordered*: there is no ordering between the steps. They may overlap or not, and each may start and end whenever appropriate. Still, like in the other three cases, each of them starts after the start of the parent plan and the parent finishes when the last one of the children finishes.

The number or part of the steps, which must be performed before the plan can complete successfully, is defined by the continuation specification given in *wait-for*. This continuation specification contains a logical expression based on the names of the child plans. There are plans that wait for all their children, one of their children or none of their children (which means all children are optional).
Apart from the continuation specification there are two other attributes influencing the execution of the subplans : *wait-for-optional-subplans* and *retry-aborted-subplans*.
In addition to the continuation specification each plan has a waiting strategy defined in the *wait-for-optional-subplans*-attribute. It is no by default which means that the plan terminates as soon as both complete condition and continuation specification evaluate to true and the earliest finishing shift is reached. If this attribute is set to yes, the parent waits for its children to complete as long as the time annotation allows, even if they are not necessitated by the continuation specification.
The normal behavior of most plans is to try each subplan once. Still in some cases it would be desirable to try a set of subplans over and over again, until one or more of them succeed. For such cases the *retry-aborted-subplans* is said to yes.
Below the "highest" plan of the diabetes protocol ***Diabetes-Mellitus-Type-2*** shows a sequential plan body, a *wait-for-optional-subplans* that is set to yes and a continuation specification to wait for only one of the two subplans:

plan Diabetes-Mellitus-Type-2
intentions
achieve overall-state: (glucose-monitoring ≠ bad)
plan-body type = sequentially,
wait-for-optional-subplans = yes
wait-for one
Diagnostics
Policy

2.4.5.3 Single step

The element single-step represents one of the following:

1. *plan-activation*: activates another plan.
2. *variable assignment*: assigns a value given by an arithmetic expression to a variable.
3. *set-context*: sets a context-variable to particular value.
4. *ask*: asks the user to enter a new value for a certain parameter.
5. *list-manipulations*: performs various list manipulations which do not return a value.
6. *if-then-else*: if the condition given evaluates to true, the first single step is taken, otherwise the second is performed.

Below an example of the jaundice-plan ***Albumin-test-manual*** in which the plan body consists of a single step in the form of an ask statement.

plan Albumin-test-manual
Albumin test
conditions
activate-mode: manual
plan-body
ask albumin-in-urine

2.4.5.4 User-performed

If a plan has a plan body consisting of the element user-performed it is designated as a plan which is executed by the user, not the system. The plan's name and arguments are displayed by the user interface and the user notifies the system about the end and outcome of the plan's execution, i.e. whether it completed or aborted. Still the conditions of the plan work as usual. So if the plan has a complete condition and if this condition is fulfilled, then the plan completes.
For example the diabetes protocol contains the following user–performed plan:

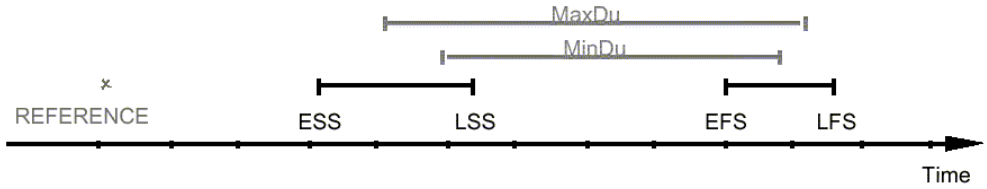
plan Funduscopy
Perform a funduscopy
plan-body user-performed

2.4.6 Time annotation

In the medical field temporal aspects play a very important role, but they often go together with a high degree of uncertainty. The time annotation in Asbru allows the representation of uncertainty in starting time, ending time, and duration of a time interval. The temporal annotation represents for each interval the earliest starting shift (ESS), the latest starting shift (LSS), the earliest finishing shift (EFS), the latest finishing shift, the minimal duration (MinDu) and the maximal duration (MaxDu). The time annotation is written as:

([ESS, LSS],[EFS,LFS],[MinDu, MaxDu], REFERENCE)

and can be graphically represented as [28]:



The general purpose of a time annotation is to constrain the temporal occurrence of plan elements (including plans themselves): the plan element has to start in a certain interval, the starting interval, and has to finish in a certain interval, the finishing interval. Finally its duration has to be within certain limits, the duration interval.
The jaundice plan ***Check-for-jaundice-after-2-weeks*** contains a filter-precondition with a time annotation, which states that jaundice should be clinically significant at least two weeks after the birth date.

conditions
filter-precondition: (jaundice-clinically-significant = yes)
in ([-,2w] [2w,-] [-,-] birth-date)

2.5 The formalization process

"There are some remedies worse than the disease." - Publilius Syrus

Formalization is a labor-intensive process by which the protocol contents are transformed into their equivalent Asbru version [22]. Sometimes the Asbru functionality requires details that otherwise wouldn't be necessary, for example the minimal and maximal delay time necessary for the specification of the retry delay in a cyclical plan. Furthermore an Asbru translation sometimes requires explicit elements, which are not explicitly part of the protocol. This is often the case with the decomposition of plans into subplans.
During the formalization process any problem is considered as a possible anomaly in the protocol, where anomaly indicates anything that prevents a satisfactory interpretation of the original protocol. In addition to this, the resulting Asbru translation is more susceptible to analysis than the original protocol and hence it helps in the identification of problematic parts [22].
The purpose of the formalization process is to uncover ambiguous, incomplete or even inconsistent parts (anomalies) of the protocol by defining all the descriptions more precisely using a formal language. During the Asbru formalization of the two selected protocols, numerous anomalies have become apparent [29]. This is a surprising result, because the two selected protocols are considered to be of high quality within the medical world. Four categories of anomalies can be recognized, namely:

Ambiguity
The problem with ambiguity is that it enables wrong interpretations by the practitioner and may therefore have a negative affect on the decisions taken.
An example of ambiguity in the jaundice protocol is the formulation of the similar terms "jaundiced" and "clinically jaundiced" in such a way that it is unclear if they should be read in connection to each other or not.

Incompleteness
Incompleteness can be related to insufficient or completely missing information. Like ambiguity it enables wrong interpretations and can have negative consequences for the final decision. Moreover it can result in an efficiency loss, when the practitioner needs to consult additional information sources to resolve the incompleteness.
An example of incompleteness in the diabetes protocol is the use of the abstract notion "rapidly rising TSB levels" without an indication of when the TSB rise is considered to be rapid.

Inconsistency
Inconsistencies are elements that may somehow result in different (and even conflicting) decisions given the same patient data. The problems caused by inconsistent elements are very serious and must be avoided.
An example of an inconsistency in the jaundice protocol is a recommendation considering children in their first day of life: "A TSB level should be determined in infants noted to be jaundiced in the first 24 hours of life". Yet at the same time the protocol considers these children to be not healthy and are therefore out of the scope of the protocol.

Redundancy
Whenever a part of the protocol can be removed without any (noticeable) effect in the resulting prescriptions, it can be considered redundant. The main problem of redundancy is that it may cause unnecessary inconvenience for the patient and efficiency loss in the care process.
An example of a redundancy discovered in the diabetes protocol is the repeated question for a patient's weight during the annual control.

However the translation of medical protocols to their formal representation in Asbru is not an easy task:

- in practice it turn out to be a very time-consuming task, also depending on the complexity of the original protocol, that can be shortened by practice and examples.
- Asbru is a complex language and learning it is difficult since there are not enough good resources available yet.
- it isn't possible to translate the protocol in Asbru in one step, which means intermediate models are necessary.
- medical protocols are written for people having medical background knowledge. During the formalization the textual version of the protocol has to be followed, but meanwhile also common medical knowledge has to be added, in other words the modeler has to understand the protocol in medical sense. So co-operation with and feedback from medical experts is crucial.
- throughout the modeling process a lot of modeling decisions about how and what to model have to be made by the knowledge engineer. Because of the lack of modeling directives this is still difficult.
- the final translation in Asbru is hard to understand by medical practitioners.

This is why there is a need for tools to assist the formalization process. One kind of such a tool that has recently been developed is the Guideline Markup Tool and will be discussed in the next chapter.

2.6 The Guideline Markup Tool (GMT)

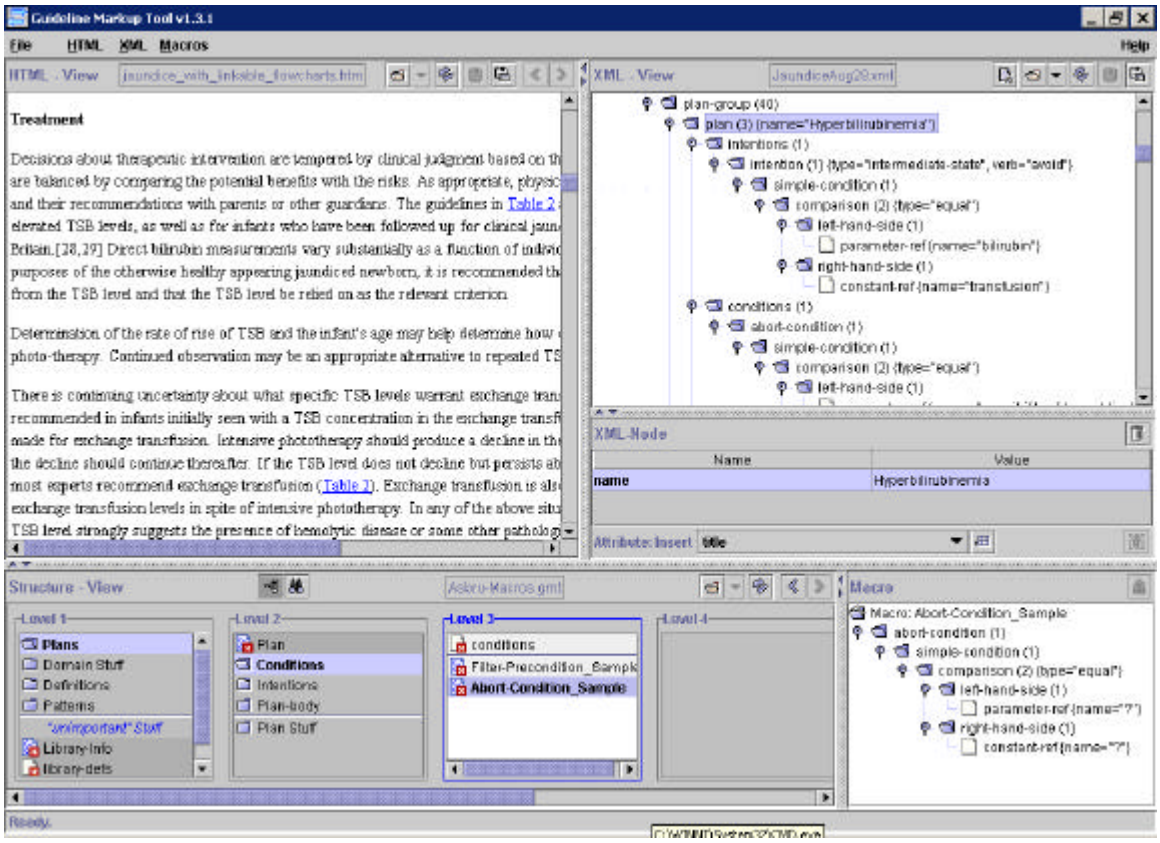
"By medicine life may be prolonged, yet death will seize the doctor too."- William Shakespeare

Within the Asgaard project currently the Guideline Markup Tool (GMT) is being developed. This is an editor that among others helps translating guidelines from free text into Asbru [10]. It has to support the following tasks [19]:

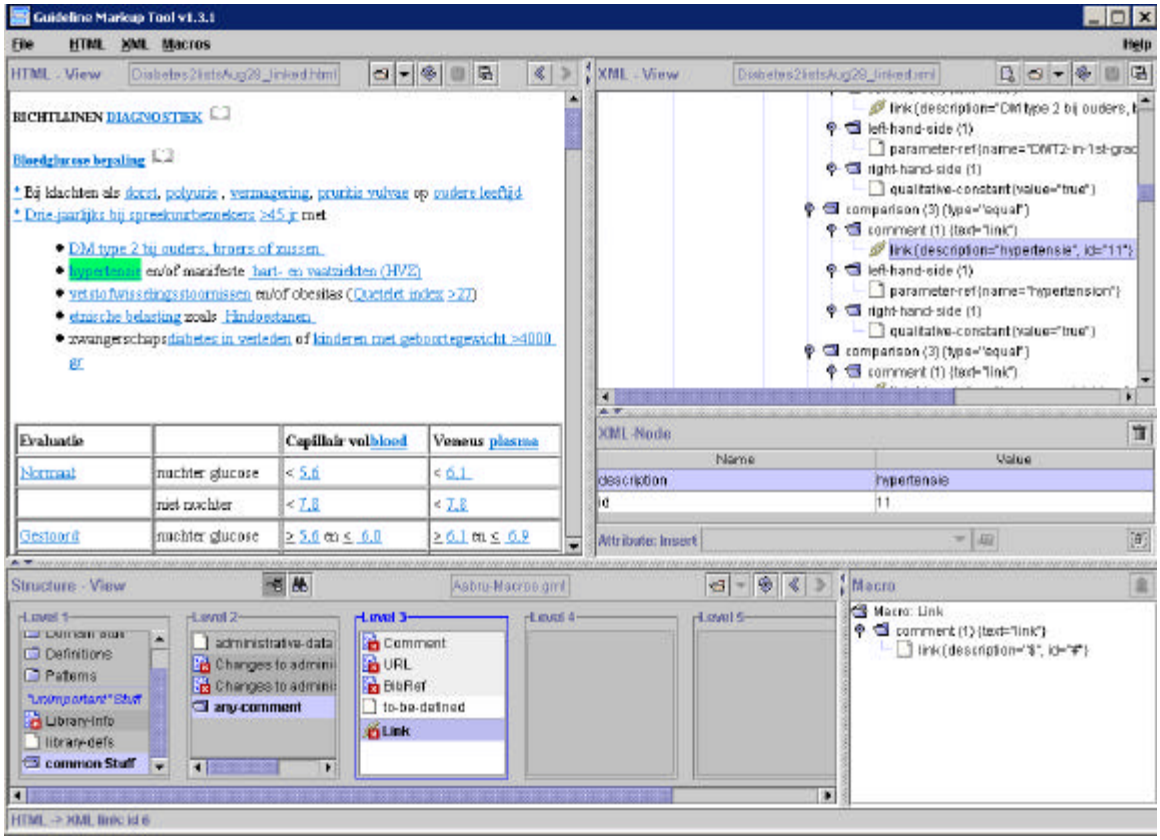
- *Authoring guidelines*: taking a new guideline in plain text and translating it into an Asbru version.
- *Understanding guidelines*: to see where values in the different parts of the Asbru code come from, and how parts of the original were translated into Asbru. This is important not just for knowledge engineers, but also for physicians wanting to get an understanding of the language.
- *Structuring Asbru*: the GMT provides a structured list of Asbru elements, called macros, that needs to be done in a way that best supports the authoring of plans.

The GMT window consists of three parts, which can be seen below:

- *the HTML View* on the left side simply contains the original guideline, where the user can select parts to be connected to Asbru elements.
- *the XML View* on the right side shows the Asbru XML tree, with the deeper nested parts giving a more detailed view of the currently selected element (with all its attributes).
- *the Structure View* on the bottom contains all Asbru elements as well as macros that the user can insert. These elements are organized in a tree, which is represented by the list boxes: each box shows the contents of the node selected in the box to its left. Once a leaf is selected, its contents (the macro or single tag), is shown in the “Macro” window on the bottom right.



An utilization of the Guideline Markup Tool would be to open an original protocol in the HTML View, a file containing Asbru elements in the Structure View and then construct the Asbru translation of the protocol in the XML View by using the elements from the Structure View.
Another functionality of the Guideline Markup Tool, which has been mainly used in this graduation project, is the link macro. This macro enables the definition of link between a piece of HTML text from the original protocol and a XML node in the Asbru protocol. What in reality happens when a link macro is inserted is that the selected piece of HTML text as well as the selected XML node both are given the same identification number and links can be seen as connections between HTML and XML nodes with corresponding identification numbers.
For using the linking functionality first the original protocol is opened in the HTML view, its Asbru translation in the XML view and a file containing the link macro in the Structure View. Next the links between the original protocol and the Asbru translation can be inserted with the link macro. Once the links are all defined a better insight in how the two protocols are related is obtained.
The screenshot below shows an example of a link between the original diabetes protocol and its Asbru translation. The link considers the relation between the examination for hypertension during the triennial visit and its translation in Asbru that checks if the parameter hypertension is true. Links are underlined in blue on the HTML side and represented as a link node on the XML side, which is placed under the node it belongs to. If the user clicks in either the HTML view or the XML view on a link, the accompanying link on the other side is highlighted in green.



3 Linking with the Guideline Markup Tool

The result of the formalization process is an Asbru protocol, which exists next to the original protocol. Medical practitioners only understand the original protocol and except for the modelers nobody really has insight in the relation between the original protocol and its formalization. That is why there is a need for a way to show this relation between the original protocol and its formalization.
Recently the Guideline Markup Tool has been developed as an editor that helps translating guidelines from text into Asbru. This tool also has the functionality to define links between HTML and XML. So with this tool all the relations between the original protocol in HTML and its formalization in XML can be defined in the form of a link.
These links between the original protocol and its Asbru translation in GMT can serve different purposes:

- to give insight in the relation between the original protocol and its formalization. For example in the big size difference between an original protocol and its Asbru translation. This bigger complexity of Asbru protocols will be discussed separately in chapter 5.
- to enable discussion with domain experts. The domain expert can refer to the original protocol he understands and with the help of the links the knowledge engineer can see which part of the Asbru protocol the domain expert is talking about.
- to reveal if everything in the original protocol that should have been modeled also really has been modeled.
- in case the original protocol is updated changes can much easier be made in the Asbru translation, since the place where the adjustment should be made can easily be found with the link pointing there.
- to help the modeler during the formalization process. By thinking about the links already during the formalization process, the modeler has to consider the consequences of his modeling choices on the relation between the original protocol and its Asbru translation.

Within this graduation project this functionality has been used to define all the links between both the jaundice and diabetes protocol and their Asbru translations. This chapter will discuss this linking with GMT. First the linking task itself will be described and then the outcomes of the linking of the two protocols.

3.1 The linking task

As already explained in paragraph 2.6 about the Guideline Markup Tool a link can be defined by selecting a piece of HTML text from the original protocol, a XML node from the Asbru protocol and then inserting the link macro. But how do you know what piece of HTML text and which XML node to select? In other words: where do you put the links?
First of all the linking task starts with a thorough investigation of both the original protocol and its Asbru translation, because to determine if there is a link or not one needs to know exactly what happens. This investigation can already give rise to a lot of questions.
In the best situation the knowledge engineer that has done the translation to Asbru is still available for questions, which was the case for this graduation project. If this isn't possible however other sources of information have to be sought. For the original protocol medical sources can be consulted and for the Asbru protocol perhaps knowledge engineers that have translated other protocols to Asbru can be a helping hand.
Once a good understanding of the protocols has been obtained the linking can start. Like the modeler has to decide whether and how to model something in Asbru, the linker has to decide whether and how to link something. Some aspects of the original protocol are very obviously

related to the Asbru translation, most of the time however the relation is less obvious. In this case the linker has to think about the best way to define the link.
As already mentioned for this graduation project the links between both the jaundice and diabetes protocol and their Asbru translations have been defined. This resulted in 253 links for jaundice and 276 links for diabetes. It should be noted that these numbers only represent the unique link identification numbers and so do not include multiple links (multiple links get the same identification number). So in practice the protocols even contain more links.
The links that have been defined for the two protocols can roughly be distinguished in two ways. A link can be characterized as explicit or implicit. Besides the level (high or low) in Asbru at which a link is defined can be different. These two distinctions are explained in more detail below.

3.1.1 Explicit and implicit links

One could say two types of links can be identified, namely explicit links and implicit links. Explicit links are links that show a very direct, obvious relation. The following example from diabetes and jaundice respectively show an explicit relation:

Original		***Asbru***
"fat-metabolism-problems"	→	if (or(..)(fat-metabolism-problems = true))
"Does this infant have abnormal physical exam, dark urine, or light stool?"	→	if(or(physical-exam-OK = no)(…))

Implicit links on the other hand are much less apparent. They do relate two parts of the original and formalized protocol that belong together, but the relation is not completely fitting. There are several reasons for implicit links.
First of all, domain experts may have clarified terms that are a bit vague in the original protocol. This results in a detailed statement in the formalization, which is related to a more vague statement in the original protocol. An example from the diabetes protocol clarifies this:

Original		***Asbru***
"pruritis vulvae at older age"	→	**if** (and (age>60)(sex = female)) **then** **ask** pruritis-vulvae

In this example "older age" is the vague term and with advice of domain experts this has been translated in the Asbru protocol with "age>60".
Second of all, the value of a certain parameter may be used in the protocol, but to be able to use it, the value first needs to be obtained. Original protocols mostly consider it to be clear the value has to be obtained and don't mention it explicitly. In the Asbru translations on the other hand an explicit ask statement has to be used. This ask statement can be implicitly related to the place in the original protocol where the parameter value is really used. This can be illustrated with the following example from the diabetes protocol:

Original		***Asbru***
"Quetelet index > 27"	→	**ask** quetelet-index

Third of all, common knowledge in the original protocol can cause a different translation in the formalization. For example in the diabetes protocol blood pressure needs to be checked. It can be considered as a common fact that for checking blood pressure, both lower and higher blood pressure need to be measured. In practice this means the following:

Original		***Asbru***
"Blood-pressure"	→	**ask** lower-blood-pressure **ask** higher-blood-pressure

Fourth of all, sometimes related aspects in the original protocol are put together in a "superplan", which is divided in subplans that represent these different aspects. Besides explicit links to these subplans, an implicit link from the "superplan" to the collection of related aspects is desirable. Below an example from the diabetes protocol, which has three explicit links to the different forms of cholesterol tests and one implicit link from the combination of tests to the superplan *Cholesterol-tests*. This already shows one of the missing functionalities of the Guideline Markup Tool. It is either possible to make the three links separately or one composed link in GMT. This missing functionality will be discussed in more detail in the next chapter.

Original		***Asbru***
"total cholesterol[1], HDL-cholesterol[2], tryglycerids[3]"	→	**plan** Cholesterol-tests[4] *Cholesterol tests* **plan-body** type = sequentially
"total cholesterol, HDL-cholesterol, tryglycerids[4]"	→	**wait-for** all **ask** total-cholesterol[1] **ask** HDL-cholesterol[2] **ask** tryglycerids[3]

These are some of the reasons for implicit links, but more can be identified. In any case the distinction between explicit and implicit links shows that some of the relations between the original protocol and its formalization are very obvious, but others are much more indirect for various reasons.

3.1.2 High level and low level links

The Asbru protocol is represented in the Guideline Markup Tool as a XML tree. The characteristics of a tree structure is that nodes appear at different levels in the tree. This means links can also appear at different XML levels depending on the XML node that has been selected for the link.
Most of the links appear at a low level, which means they can't be subdivided any further and no links will show up in their child nodes. In this case the parent node and all its child nodes belong to the link. The following example from the jaundice-plan ***Anamnesis-abnormal-signs*** illustrates this:

Original		***Asbru***
"excessive weight loss"	→	**ask** excessive-weight-loss

Here the medical practitioner should ask a patient for excessive weight loss. This ask statement appears at a leaf node in the XML representation of the Asbru protocol, so the link to this ask statement appears at the lowest possible level.

Other links however appear at a higher level. These links often cover a broader area and some of the child nodes of this link can be separately linked on a lower level as well. An example from the diabetes protocol clarifies this:

Original		***Asbru***
"make a four points day curve[5] (fasting[1] and 2 hours after the tree main meals[2,3,4])"	→	**plan** Four-points-day-curve[5] **plan-body** type = sequentially **wait-for** all
	→	**ask** fasting-glucose[1] **ask** after-breakfast-glucose[2] **ask** after-lunch-glucose[3] **ask** after-dinner-glucose[4]

In this example the statement "Four points day curve" in the original protocol can be related to the plan ***Four-points-day-curve*** in Asbru, though the subnodes of this plan also contain lower level links to the original protocol. So this link is related to the parent node only and not to the child nodes which are still separately linkable.
It should be noted though that these high level nodes mainly consider links from the original protocol to a whole plan in Asbru. It has been tried to define all the other links at the lowest level possible.

3.2 Outcomes of the linking

The linking task itself already asked for a critical attitude towards the original protocols and their Asbru translations to be able to determine how they are related. Once the linking of the protocols with GMT had been finished, the linked protocols were examined further. The results that have been obtained and the observations that have been done will be discussed below.

3.2.1 New anomalies uncovered

One of the most surprising results of the linking task is that new anomalies have been uncovered. Some hadn't been identified during the formalization process and others had even been introduced during the formalization process. The contradictory thing about this last finding is that one of the main purposes of formalization is to improve the quality of a protocol, but apparently it can also introduce new mistakes and so reduce the quality.
Concerning the anomalies that hadn't been identified during the formalization process, most of them considered typing errors. For example in the jaundice protocol in the flow diagram, there is a box with the text "From box 12", looking at the flow diagram it is clear that this should have been "From box 11".
Furthermore in the same protocol table 2 appears twice, once in the text and once in the flowchart. There are two inconsistencies between these tables. The first considering the value in column "Consider phototherapy" for age 25-48, where there are two different values, namely 210 pmol/L and 170 µmol/L. It might be already clear that the second inconsistency has to do with the differing unities in these tables pmol/L and µmol/L. These typing errors can't be considered as very serious problems, but still they make the protocol inconsistent.
As for the anomalies that have been introduced by the formalization process, most of them have been found in the diabetes protocol. One important observation about this protocol is that the original version of the protocol is written in Dutch. Since the formalization has been performed by a non-Dutch speaking knowledge engineer, the protocol first has been translated in English to be able to do the modeling.

It is this translation that has introduced new anomalies, because it was not completely consistent with the Dutch version. For example, the original protocol in Dutch says “in case of pressure-points, corn, bad standing habits”, which has been translated to “in case of pressure-points, bad standing habits caused by corn”. The last has been formalized in Asbru, but doesn’t agree with the interpretation of the original Dutch version.
Of course the linking could have also been done with the translated version and then this anomaly would have probably not been discovered. So in case translation to another language (which will be in general English) is necessary, this has to be done really accurately to be able to guarantee similarity and so to guarantee no new anomalies.
Another anomaly found in the diabetes protocol is that for “refer to podiatrist or pedicure with diabetic note” only the podiatrist has been modeled in Asbru. The reason the pedicure has been left out of consideration is the modeler considered them to indicate the same specialists. This turns out to be a wrong assumption since a podiatrist must have a medical degree and for a pedicure only a diabetic note is enough, which means they both should have been modeled.
A last anomaly in the diabetes protocol considers the quetelet index. This quantity appears twice in the original protocol; first as quetelet index > 27 during the anamnesis of people older than 45 and second as quetelet index < 27 during the treatment with antidiabetics.
The last appearing quetelet index already seems to be incomplete, since it doesn’t cover a quetelet index of 27. Perhaps this is another anomaly of the original protocol. A medical expert should be consulted to find out why this value isn't covered and so to determine if it considers an anomaly or not.
This quetelet index was translated as quetelet index ≥ 27 in the first case and as quetelet index ≤ 27 in the second case. Probably the modelers wanted to close the gap for the value 27, but didn’t notice they did this in an inconsistent way and so introduced a new anomaly.

3.2.2 Already uncovered anomalies visualized

During the formalization process of the two protocols different anomalies have been identified and documented [22]. As a result of the linking some of these anomalies now have been visualized, because they remain unlinked.
For example the jaundice protocol says “A TSB level needs to be determined in infants noted to be jaundiced in the first 24 hours of life”. At the same time the protocol considers these children to be unhealthy and so their treatment is outside the scope of the protocol (the protocol only considers the management of jaundice in the healthy term newborn). This was noted to be an inconsistency of the protocol [22] and it was decided not to represent this in Asbru. This is what also can be seen as a result of the linking: this whole sentence in the original protocol remains unlinked.
Another anomaly that was identified in the diabetes protocol considers the redundant data request for the weight in the annual control [22]. That is why in the Asbru translation only during the quarterly control the weight is asked. The linking shows that this second request for weight in the annual control has no relation with the Asbru protocol, because in the original protocol it remains unlinked.

3.2.3 Unmodeled parts of the original protocol visualized

After the linking task was finished some parts of the original protocol remained unlinked. Because all the links are underlined with blue in GMT, it gives a very good visualization, which parts of the original protocol have not been modeled in the Asbru formalization. Usually because the modeler has decided not to model something.

Of course the examples about uncovered anomalies described in paragraph 3.2.2 already illustrated parts of the original protocol that remained unlinked and so had not been modeled in Asbru.
Further the diabetes protocol contains “obesity (Quetelet index > 27)”. In the Asbru translation only this quetelet index is modeled, so “obesity” remains unlinked. The reason is that a quetelet index above 27 means obesity, so actually this can be considered as redundant information. Since quetelet index also appears in the rest of the protocol this term was chosen to be modeled.
Another example of an unmodeled part of the diabetes protocol is the sentence “2 hours after the 3 main meals” considering the glucose levels. This has been modeled with three asks for after-breakfast-glucose, after-lunch-glucose and after-dinner-glucose, which means that these 2 hours have disappeared in the formalization and can’t be linked. This modeling decision considers the doctor to be capable of doing the tests in the right way.
The examples above consider unmodeled parts of the original protocol that can be missed in the Asbru translation. An example of an unmodeled part of the original protocol that could not be missed on the other hand is the following phrase from the diabetes protocol: "When the patient is back in consciousness". By now this has been corrected and added to the Asbru translation of the diabetes protocol, but it is a good example of how the linking can show if everything from the original protocol that should have been modeled also has been modeled.
The summary version of diabetes that is used for the modeling is quite concise, which means that most of the text has been modeled in the Asbru translation. Jaundice on the other hand contains a lot of information irrelevant for the formalization, which means considerably more parts of the original protocol remain unlinked. When the longer version of diabetes would have been modeled, probably also much more parts of this original version would have remained unmodeled.
Already the first one and a half page of the jaundice protocol remains unlinked, because it contains only background information irrelevant for the diagnosis or treatment of jaundice.
Also in the other parts of the protocol that do consider diagnosis and treatment there are pieces of text that show no relation with the Asbru translation. For example the sixth item of the evaluation part says, “The clinical assessment of jaundice must be done in a well-lighted room”, which can’t be found back in the Asbru translation. This same item also contains “As the TSB level rises, the extent of cephalocauded progression may be helpful in quantifying the degree of jaundice”, which also isn’t translated in Asbru. Again this are examples of a modeler's choice not to model everything, but only the essential things.
Furthermore the protocol contains a lot of additional pages with committees, references and appendices, that don’t appear in the Asbru translation. Although a considerable amount of text remains unlinked, this doesn’t apply for the tables and flowchart contained in this protocol. They do contain quite a lot of relations with the original protocol.

3.2.4 Unlinked XML parts uncovered

As well as there are parts of the original protocol that remain unlinked, there are even more parts of the Asbru translation that remain unlinked, because there is no direct relation with the original protocol. The visualization of unlinked parts of the Asbru protocol in the XML View is less apparent than for the underlined parts of the original protocol in the HTML View. That is why a visualization functionality, discussed in more detail in the next chapter, has been added to the Guideline Markup Tool that distinguishes linked and unlinked XML parts by color. With this new functionality it can be very easily seen which XML nodes do or don't contain a link.
All kind of unlinked parts can be distinguished, but mostly all of them indicate increased complexity to the original protocol. For now this won't be discussed any further, since the increased complexity of Asbru protocols will be discussed as a separate topic in chapter 5, which will cover the unlinked XML parts.

3.2.5 General observations

Besides the outcomes described above some more observations about the protocols have been done that are worth mentioning. Below these additional observations are described for each of the two protocols.

Jaundice

1. The plan ***Diagnostics-and-Treatment-hyperbilirubinemia*** contains two abort conditions that correspond to boxes 4 and 16 in the flowchart. By these abort conditions the steps in the Asbru protocol are done in a different order than in the original protocol, because if these conditions are fulfilled boxes 4 and 16 will be executed before the other boxes in the flowchart. The overall effect will be the same, but still it means the protocol is not exactly modeled according to the original protocol.
2. The original jaundice protocol contains a lot of redundancy. A lot of multiple links have been defined to relate these redundant parts, like for example the (identical) reappearing tables, to the Asbru protocol.
3. The original protocol is not clear about if maternal testing belongs to the protocol or not. On one hand the protocol states "maternal testing should include…", implying maternal testing should be done, but on the other hand the flowchart advises to perform a blood test with the child if there is no blood test of the mother available, implying that no maternal testing has to be done.
4. The part of the Asbru protocol representing the table considering when to start phototherapy or exchange transfusion where difficult to link with the original protocol. The relations with the original protocol are not completely evident. So decisions had to made how to define these links in a way the relationships are represented best.

Diabetes

1. Diabetes consists of a long and a short summary version. This summary version has been modeled in Asbru. Looking at the longer version though, it can be seen this is not covered by this Asbru translation. For example the list of typical signs is much longer than in the summary version. The question is whether the summary version covers the diagnosis and treatment of diabetes enough or some aspects only mentioned in the long version also should have been modeled.
2. The original protocol says "albumin concentration or albumin/creatin ratio in the first morning urine". This can be interpreted in two ways: either both values are measured in urine or only the second is. The modelers have chosen to measure both in urine, but from the text of the original protocol this is not 100% clear.
3. The Asbru protocol models both non-fasting and postprandial (after meal) glucose. The determination of postprandial glucose should always be a certain time after a meal, while for the determination of non-fasting glucose no care has to be taken about fasting. So these terms do show a certain overlap, but it is not clear from the protocol why to use postprandial in one case and non-fasting in the other. Perhaps only using postprandial values in all cases could also suffice.

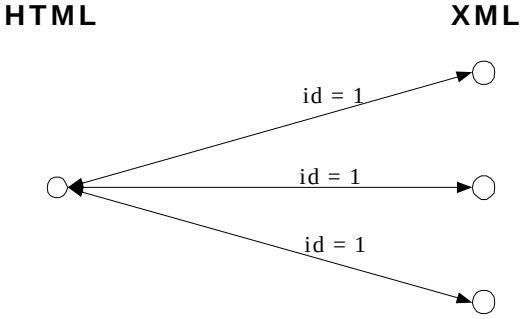
4 Evaluation of the Guideline Markup Tool

As described in paragraph 2.6 the Guideline Markup Tool is an editor that helps translating guidelines into Asbru. An additional functionality of the tool is to define relations between the original protocol and its Asbru translation with a link macro. The status of the Guideline Markup Tool at the moment is that all the important features have been implemented. Yet there is still a list of things to be implemented, mainly to improve user convenience.
A logical result of being the first user of the tool was performing an evaluation of the Guideline Markup Tool. Of course this included trouble shooting of bugs that had not been found during testing and made normal execution of the program impossible. Yet since the Guideline Markup Tool has particularly been used for its linking functionality during this project, the evaluation mainly focused on the use of this functionality. So does the tool completely support the linking task or are there things that can be improved? This chapter will describe the observations that have been made during the evaluation.
In communication with the developer of the tool meanwhile a lot of bugs have been resolved. Furthermore some new functionality to improve the linking task has been added, which will be described in the paragraph “added functionality”. Still there is some functionality that is essential for the linking task not implemented yet. This will be described in the paragraph “missing functionality”. Finally there is some functionality not implemented that would make the linking task much more convenient. It is not absolutely necessary, but would be a nice addition. It is described in the paragraph “desirable functionality”.

4.1 Added functionality

4.1.1 Multiple links to XML

Defining links in GMT from one place in HTML to different places in XML seemed to be possible at first sight, but it turned out that previously made links were overwritten by the new links. After an adjustment of GMT this problem has been solved, so now it is possible to define links from one place in HTML to several places in XML, which all get the same identification number.
As explained in paragraph 2.6 about GMT what actually happens when a link is defined is the two selected nodes in HTML and XML both get the same identification number and links can be seen as connections between HTML and XML nodes with corresponding identification numbers. For multiple links this means when an HTML node has a certain identification number all its related XML nodes share the same identification number.
Below a graphical representation of multiple links from one piece of HTML to three different nodes in XML is shown:



For example in the diabetes protocol there are two places in XML that are related to “thirst” in the original protocol:

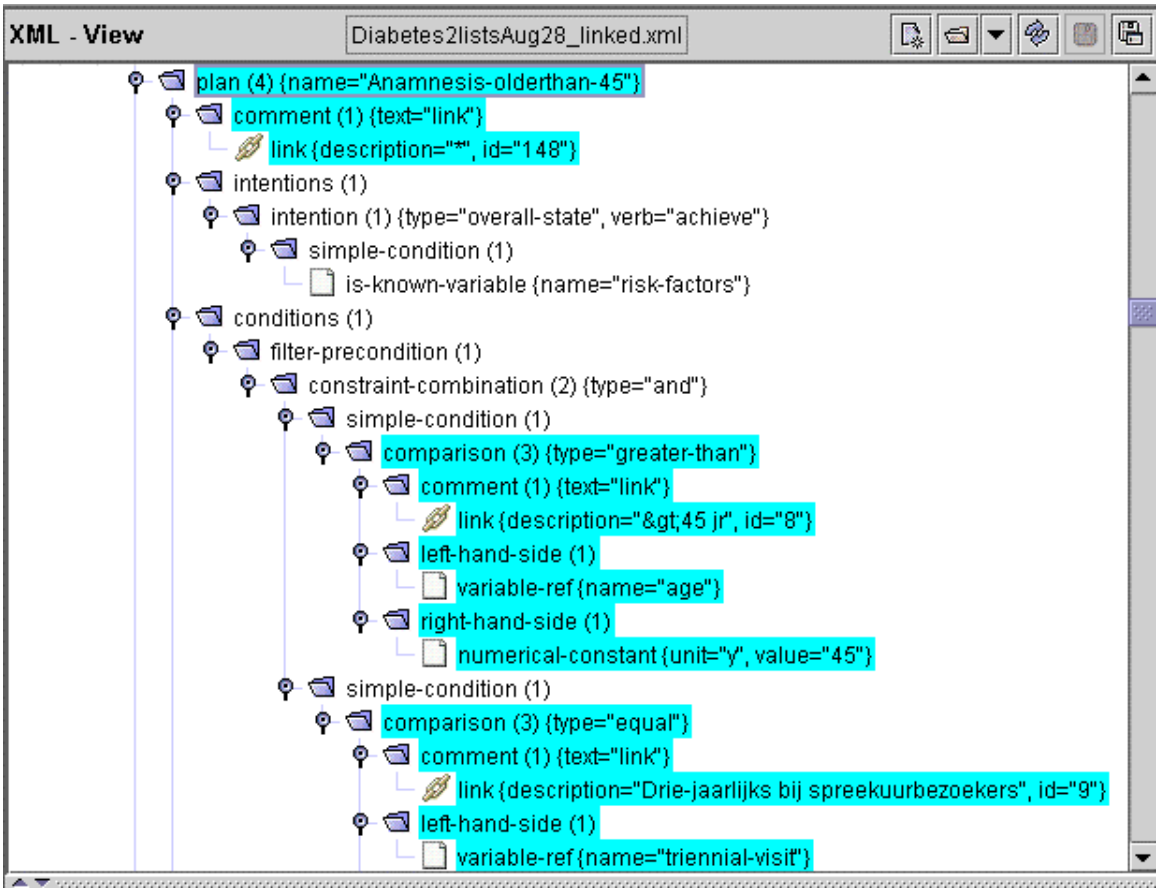
Original		***Asbru***
"thirst”	→	**ask** thirst
	→	**if** (**or** (thirst = true) (…))

4.1.2 Scrolling through multiple links

Once the functionality of multiple links was added to GMT, it was possible to define different links from one place in HTML to several places in XML. The only thing was that selecting this HTML link in the HTML view always showed the first of the links in the XML screen (the link is highlighted in green). Yet to be able to see the other links belonging to this HTML link, the XML document had to be scrolled manually to find the other green-colored links.
This has been improved now, since with each click on an HTML link, the program jumps to the next link in the XML screen, which makes it much easier to scroll through multiple links.

4.1.3 Visualization of linked and unlinked XML parts

One of the aims of linking the protocols is to see which parts of the original and Asbru protocol show no relation with the other side. For the HTML view this can be seen at one glance, since the links are underlined with blue. For XML on the other hand this was much less apparent.
The person that has done the linking has this information in its head. Yet for an outsider the only way to know which parts of the Asbru formalization are not linked, is to go through all the nodes and determine which do or don’t contain a link. Since this isn’t really convenient, a way to visualize the distinction between linked and unlinked XML nodes was being sought.
A visualization option for the XML view has now been added to GMT. It distinguishes nodes that belong to a link from the ones that don’t by giving them different colors. The only problem in developing this visualization functionality was the fact that different kinds of links exist that have to be colored in a different way to represent the coloring right.
The distinction between the different links that has to be made for the right visualization has to do with the level on which the links appear. As already explained in paragraph 3.1.2 most of the links appear at a low level, which means they can’t be divided any further, in other words that no other links can appear in the child nodes. In this case the parent node of the link and all its child nodes belong to the link and so should be colored.
Some of the links however appear at a high XML level, covering a broad area, which means some of the child nodes can be separately linked on a lower level as well. In this case only the parent node of a link should be colored and the child nodes are possibly linked and so colored on a lower level or not. For example a link to an entire Asbru plan can often be considered as a high level link, since some of the child nodes of the plan can also be related to the original protocol on a lower level. It would make no sense to color the whole plan in this case, since there are always parts of a plan, like for example intentions, that show no relation with the original protocol and so would be erroneously colored.
Below an example of the visualization of the links in the diabetes protocol is shown. It can be seen that for the link belonging to the plan node only its parent has been colored. For the link belonging to the comparison on the other hand, both the parent node and all its children have been colored, since they are all covered by the link.



4.1.4 Editing links

When the user defines a link, GMT automatically picks the selected HTML text as link description and automatically assigns a subsequent identification number to the link. At first the link description and identification number were unchangeable once they were set, but now it is possible to edit both the identification number as well as the description of a link. This makes it possible to give a link the desired description and to change the link identification easily without having to delete a link first and then redefine it.

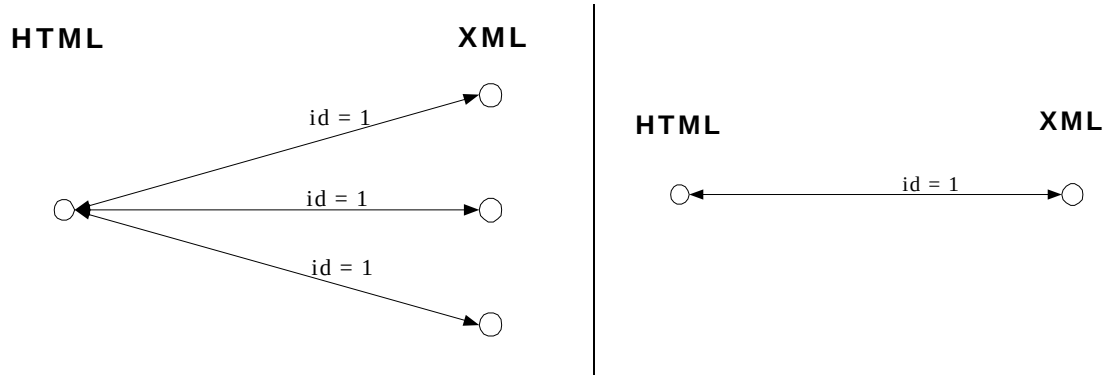
4.1.5 Confirmation of node deletion

GMT has a button to delete XML nodes, which was executed as soon as the button was pressed, without any request for confirmation. So if the button was erroneously pressed a node (which could mean a whole plan) would be removed and since the undo option also hasn't been implemented yet, this could have very unpleasant outcomes.
Luckily this inconvenience has been solved. Now each time the delete button it pressed, the user first has to confirm the deletion. So if the button is pressed erroneously, there is still a possibility to cancel.

4.1.6 Deletion of dead links

GMT constructs links by giving both an HTML node (a selected piece of text) as well as a XML node the same identification number. So actually links can be seen as connections between all HTML and XML nodes with corresponding identification numbers.

Links can be deleted either from the HTML view or from the XML view by first selecting the link and then selecting the remove option. If there are no more link nodes with the same identification number in the selected view, this means that the link nodes with this identification number on the other side become “dead” (pointing to nothing since there are no related nodes with the same identification number anymore) by the deletion. In this case GMT asks the user if the dead links should be removed and if the user confirms does so.
This new functionality was needed by the introduction of multiple links. The introduction of multiple links made several links with the same identification number possible. At the same time this meant that deleting a link node from one view does not always mean the related link nodes in the other view should be removed, because they could still cover other links.
Below two examples illustrating the different possibilities when deleting a link are graphically shown:



In the example on the left if the only node on the HTML side is deleted, three dead links on the XML side arise and the user will be asked if he wants them to be deleted. On the other hand if the top node on the XML side is removed, the corresponding node on the HTML side will still cover the remaining two links, so only the XML node is removed.
In the example on the right if the link node either on the HTML or XML side is deleted, in both cases a dead link on the other side arises, so the user will be asked if he wants it to be removed.

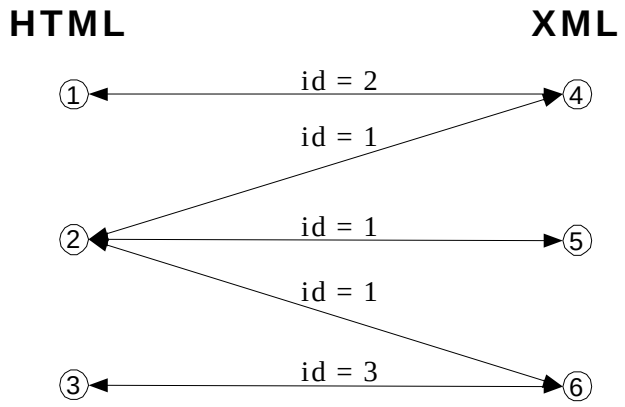
4.2 Missing functionality

4.2.1 Linking to pictures in HTML

If the protocol contains pictures, like flowcharts in image-format, no links can be made to parts of these pictures, which can be very disadvantageous for the linking task. In the jaundice protocol for example the flowcharts play an important role and show a strong relation with the Asbru translation so being able to define links to these flowcharts is desirable.
For now making an imitation of the used flowcharts in another format has solved the problem. This also wasn’t an easy task, since the HTML View of GMT is quite basic, so it can’t handle any complex HTML formats. Finally the flowcharts have been reproduced with nested tables, which turned out to be usable.
It is possible to reproduce flowcharts in another usable format, but for real pictures that are related to pieces of XML this can be a bigger problem. So in the future it is essential to find a better solution for this problem.

4.2.2 Multiple links to HTML

As explained earlier in the paragraph about multiple links to XML it is possible to define links, which all get the same identification number, from one place in HTML to different nodes in XML. The other way around it is sometimes desirable to define multiple links from one place in XML to different places in HTML, but for now this is not possible in GMT.
As explained earlier links can be seen as connections between nodes with corresponding identification numbers. It is possible to define links from one place in XML to different places in HTML, but then all these links get the same identification, which is often not desired. The different HTML nodes do not always cover exactly the same things, so the links should stay separately identifiable which can only be accomplished by giving them and the related XML node differing identification numbers.
Besides if an HTML node already covers multiple links to different XML nodes and one of these XML nodes is related to another HTML node under the same identification this can have very unpleasant outcomes. This new HTML node can suddenly become related to all the other XML nodes, because they share the same identification number.
Below an example of a possible situation for linked protocols is shown which should illustrate the problem with the multiple links to HTML. The example shows three nodes on the HTML side and three nodes on the XML side. Node 2 on the HTML side is related to all the three nodes on the XML side, so multiple links have been defined that all have the identification number 1. HTML node 1 on the other hand is only related to XML node 4. For defining the relation between the two nodes 1 and 4 a new link with the new identification number 2 has to be defined. This link can't be related to the already existing link to XML node 4 since this would mean that HTML node 1 suddenly also becomes related to XML nodes 5 and 6, because of the corresponding identification numbers.



For now this problem has been solved by defining all the multiple links from one place in XML to different places in HTML separately which can result in a long list of links below each other, which doesn't look very nice. Besides it is not possible to select a certain XML node and see all the related places in the original protocol, as it is possible to select an HTML node and scroll through all the nodes in XML it is related to. All the different links belonging to a XML node have to be searched through to see all the related HTML nodes.
A solution could be the possibility for a XML link node to contain different identification numbers, so links can just be added to an already existing "link node".

4.2.3 Composed links

Like described in the discussion about explicit and implicit links in chapter 3, related aspects in the original protocol can be put together in an overlapping "superplan", which is divided in subplans that represent these different aspects. Besides explicit links to these subplans, an implicit link from the collection of related aspects to the "superplan" is desirable.
At the moment it is only possible to link either one of the possibilities. If the user chooses to link the superplan to the collection of related aspects the whole text becomes underlined and it is not possible to distinguish different elements inside this text anymore, so not possible to define the explicit links to the separate aspects. On the other hand if the user starts with linking to the subplans, afterwards it isn't possible to define one overlapping link to the superplan anymore.
An example from the diabetes protocol, which should have three explicit links to the different forms of cholesterol tests and one implicit link from the collection of tests to the overlapping plan *Cholesterol-tests*, shows the two possibilities:

Original		***Asbru***
1. "total cholesterol[1], HDL-cholesterol[2], tryglycerids[3]"	→	**plan** Cholesterol-tests[4] *Cholesterol tests* **plan-body** type = sequentially
2. "total cholesterol, HDL-cholesterol, tryglycerids[4]"	→	**wait-for** all **ask** total-cholesterol[1] **ask** HDL-cholesterol[2] **ask** tryglycerids[3]

For now this has been solved by defining multiple links from the different aspects in HTML to both the corresponding subplan as well as the overlapping superplan. Of course this means a lot more links than necessary have to be defined and it becomes much less transparent to see in one glance to what in the original protocol the overlapping superplan is related. So functionality, in which both links to the different subplans as well as an overlapping link to the superplan can be defined, would be an essential addition to GMT.

4.3 Desirable functionality

4.3.1 Editing in the HTML view

Sometimes the way things are represented in the original protocol make it difficult to link them in the way one would want to. For example in the diabetes protocol the HTML list item is used to structure certain topics. Imagine that the user wants to link everything belonging to one list item to a certain node in XML. It is not possible to select the list dot and link it, so the only way would be to select all the text belonging to this item and link it. The only problem here is that it is not always desired to put the whole text in one link (which makes it as described above impossible to link to separate elements of this text). A way to solve this problem is to change the list manually and replace the dot icon by a textual star that can be linked.
Since editing in the HTML view isn't possible at the moment the document first has to be changed in another text editor. It would be much easier for the person in charge of the linking if this could be done in GMT immediately.

4.3.2 PDF view for the Asbru translation

Using GMT the user can view the Asbru formalization in two XML formats: source view and tree view. There already is a big difference in readability between those two views. The tree view, structured by nodes, is much easier to read and search compared to the source view. Still it is not really practical.
Another format that can be extracted from an XML protocol is a structured PDF format, also used as representation for the Asbru examples in this thesis. This format is even more readable and really useful as a source for the linking task. So a PDF view next to the already existing tree and source view would simplify the linking task.

4.3.3 Find function in XML

Working with big XML files like the Protocure protocols is a bit inconvenient in the sense that no find function exists yet. This means that an XML document has to be scrolled manually to find what one is looking for. Especially for the linking it would be very useful if the user can search for a certain term in the XML view.

4.3.4 Relating corresponding Asbru elements

Often the same parameter, variable or plan, etc. may show up at different places in the Asbru protocol. For example parameters are all defined at the beginning of the protocol and used somewhere later on in the protocol.
These reappearing Asbru elements are often related and finally end up in multiple links in GMT. With respect to the linking task it would therefore be very interesting when these corresponding Asbru elements would be related to each other in the XML view so they can be easily searched through. In this way it can be easily seen where else a certain element is used in the protocol, which will probably simplify the identification of multiple links.

4.3.5 Distinction between implicit and explicit links

As described in the previous chapter a distinction can be made between explicit links, which means there is a very direct relation, and implicit links, which means the relation is less obvious. A nice addition to GMT would be the possibility to indicate for every link if it concerns an implicit or an explicit link, for example in the form of an extra attribute (of course the possibility to add extra attributes to a link could also be used for other purposes).
Then the earlier described visualization option could perhaps be used to highlight either all the implicit nodes or all the explicit nodes only. This could give a better insight in how these different sorts of links are distributed over the protocol.

5 Complexity of Asbru protocols

Once a medical protocol has been translated in Asbru it turns out to be much more extensive than its original version. Considering the two protocols that have been formalized for the Protocure project jaundice consisted originally of 8 pages and its formalization has become 40 pages long. The diabetes translation even is 56 pages long while its original covered only 4 pages.
Compared to the original protocols Asbru translations contain a lot of elements that show no direct relation with the original protocol. In this sense the Asbru translations are more complex than the original protocols and that is why the term (increased) complexity is used to indicate the size explosion of Asbru translations.
Apparently there are some aspects related to the translation in Asbru that are responsible for this explosion. Already some suspicions about the causes existed, but they had never been really demonstrated. Defining the relations (links) between an original protocol and its Asbru translation was supposed to give a better insight in the reasons for this increased complexity. In practice the linking appeared to untangle different aspects affecting the complexity. These aspects will be discussed separately in the paragraphs below.

5.1 Implicit knowledge

Medical protocols in general don't fulfill the closed world assumption, which states that all necessary information about a problem domain is available, so that it gives a complete description of the world. Protocols often cover implicit knowledge, knowledge that is not explicitly described in the protocol, but considered to be known by the medical practitioners using the protocol. Yet for the translation of a protocol in Asbru this knowledge needs to be modeled explicitly. This is one of the reasons for the increased complexity of Asbru protocols compared to the original protocol.
For example in the diabetes protocol there are two kind of disorders only appearing in women, namely pregnancy diabetes and pruritis vulvae. The original protocol doesn't explicitly state these disorders only appear in women, but in the Asbru translation a condition (sex = female) has been added.
Using the GMT visualization option that distinguishes linked and unlinked XML nodes it can be seen that the nodes representing implicit knowledge like the condition (sex = female) above indeed remain unlinked and so have no relation with the original protocol.
An interesting remark about implicit knowledge is whether a knowledge engineer that is translating a protocol to Asbru will model it in the right way. A knowledge engineer is supposed to model what is in the protocol, but he is not always having the necessary background knowledge. Of course medical experts can be consulted in case of doubts, but implicit information also might stay unnoticed and so unmodeled.
In case of pregnancy diabetes for example it can be considered common knowledge that only women can become pregnant. Yet other situations are thinkable in which a medical practitioner is familiar with the fact a certain disease only shows up in men and is carried by women, but the knowledge engineer will not be familiar with this information and so probably won't model it.

5.2 Extra information from domain experts

During the formalization of the jaundice and diabetes protocol, in different cases additional information has been gained from domain experts. Of course modeling this extra information in Asbru causes extra complexity compared to the original protocol.
It turns out that most of the additional information is finally modeled in retry delays, conditions and intentions, which are discussed as separate complexity issues in the following paragraphs.

5.3 Conditions

Conditions control the sequence of proposed actions in the guideline. They need to hold at particular plan steps to induce a particular state transition. This Asbru component can follow from something that is explicitly stated in the original protocol, implicitly derived from the original protocol or derived from additional information that has been gained from domain experts. In the last two cases it means the conditions add extra complexity compared to the original protocol.
An example of a condition that is explicitly stated in the original protocol and so causes no extra complexity, is the following filter-precondition from the diabetes-plan ***Anamnesis-older-than-45***:

```
conditions
  filter-precondition: (and (age > 45) (triennial-visit = true))
```

This condition is also explicitly stated in the original protocol as: “during triennial visit for visitors older than 45”.
On the other hand a condition that has been implicitly derived from the original protocol and so causes extra complexity, is the following complete-condition in the diabetes-plan ***Treatments-and-Controls***:

```
conditions
   complete-condition: (alive = false)
```

It is clear that as soon as a patient dies, the treatment is stopped, but this does not have to be stated explicitly in the original protocol.
Another example of a condition that is not explicitly stated in the original protocol, which causes extra complexity, is the filter-condition in the diabetes-plan ***Smoking-advice***:

```
conditions
   filter-precondition: (smoking = true)
```

It will be evident to the reader why this condition has been added to the plan.
Both the jaundice and the diabetes protocol contain a lot of different conditions that sometimes do stem from the original protocol and sometimes don’t. Though there is one type of condition that appears in these protocols that has never been linked with the original protocol, namely:

```
conditions
   activate-mode: manual
```

This can be explained by the fact that this manual activation mode is mostly used to be able to represent a manual choice between several different plans by the medical practitioner, something that is never explicitly stated as such in the original protocol.

5.4 Intentions

When actions are performed with respect to medical diagnosis or treatment, it is often useful to realize why this action is being performed after all. Asbru offers the possibility to define so-called intentions, which express this aim of a plan. On one hand states and actions that should be true during the execution of a plan can be identified and on the other hand states and actions that should be true after the execution of a plan.

In most cases intentions aren't explicitly stated in the original protocol, but considered to be known by the medical practitioner. This means modeling intentions mostly causes extra complexity, what can be observed by looking at the linked protocols. This shows that none of the intentions in the Asbru translations are related to the original protocol.
Most of the information captured in the intentions of the Protocure protocols has been obtained from domain experts. An example of such an intention that has been obtained from domain experts comes from the diabetes-plan ***Treatments-and-Controls***:

intentions
avoid-non-desired-situations: **avoid intermediate-state**:
(**or** (hypoglycemic-coma = true) (complications = true))

This intention means the aim of treating and controlling a diabetes patient is to avoid that a patient falls into a hypoglycemic coma or that complications occur, also described as non-desired situations. Though the original protocol doesn't state this anywhere.
So intentions in Asbru provide a very useful functionality to express the motive behind a certain medical action, but one does have to pay the price of extra complexity.

5.5 Retry delays of cyclical plans

The plan body is a set of plans that can be executed in different orders. One of them is the cyclical order, which means a plan is retried on a certain time basis. In case a cyclical plan is used, the syntax of Asbru enforces the definition of the time-interval on which the plan should be repeated, the so-called retry delay. Sometimes this retry delay results explicitly from the original protocol, sometimes implicitly and sometimes it has to be gained from a medical expert. In the last two cases it means Asbru requires something not explicitly stated in the original protocol to be explicitly modeled. A typical example of increasing complexity.
An example of a retry delay that causes no extra complexity since it is explicitly stated in the original protocol can be found in the diabetes-plan ***Find-antidiabetic-dose*** for the determination of the antidiabetic dose during SU derivative or metformin treatment. In the original protocol it says the dose should be increased every 2-4 weeks (till the desired value has been reached). So in Asbru this has been translated with a retry-delay of [2 weeks, 4 weeks]:

cyclical-plan
drug-doses, antidiabetic-maximal-doses ←
Increase-drug-doses (drugs, iterator-drugs, drugs-doses, iterator-drug-doses)
r**etry-delay**: min = 2w, max = 4w

On the other hand an example of a retry delay that does cause extra complexity, because it implicitly follows from the original protocol is the annual control in the diabetes-plan ***Treatments-and-Controls***. This annual control is logically translated in a cyclical plan with a retry-delay of [46 weeks, 50 weeks]:

cyclical-plan
Annual-control
retry-delay: min = 46w, max = 50w

Interesting to note is that the jaundice protocol only contains one cyclical plan in contrast to the diabetes protocol, which contains seven of them. So apparently the characteristics of a protocol can influence the amount of cyclical plans used and so the complexity of an Asbru translation.

5.6 Conversion of parameters to variables

Executing an ask statement in Asbru returns a parameter value. Sometimes the value of this parameter is needed as a variable later on in the plan. In this case the parameter first has to be converted to a variable to be able to use it. This can for example be done with an if-then-else statement giving the new variable the value of the parameter. Since in original protocols only one entity is used for both the value represented by the variable and the parameter this is another example of increased complexity compared to the original protocol.
An example is the following conversion in the jaundice-plan ***Anamnesis-hemolytic-disease***:

```
ask par-possibility-of-hemolytic-disease
if (par-possibility-of-hemolytic-disease = yes)
then
        possibility-of-hemolytic-disease ← yes
else
        possibility-of-hemolytic-disease ← no
```

Looking at the plan ***Anamnesis-hemolytic-disease*** one would intuitively think this if-then-else statement might be unnecessary if only in the intentions *is-known-variable (possibility-of-hemolytic-disease)* would be replaced by *is-known-parameter (par-possibility-of-hemolytic-disease)*. Though having a good look at the protocol shows that the plan ***Check-for-rapid-TSB-increase*** also uses the variable *possibility-of-hemolytic-disease*, but in this case a parameter is not usable, so the conversion to a variable is really needed.
As can be seen in the example from the diabetes-plan ***Change-to-only-insulin-treatment*** below, the conversion is less complex if the parameter considers an amount instead of a Boolean as in the example above, but still it is more complex than no conversion at all.

```
ask par-morning-insulin-dose
morning-dose ← par-morning-insulin-dose
```

A solution to this conversion problem would be to abolish the distinction between parameters and variables and make one unity that represents both. In this way the extra complexity of the conversion would be resolved. Of course it will need consideration if it is possible to combine the two unities in the first place and then how to model it.

5.7 Manually vs. automatically activated plans

If the protocol enforces a manual selection by the practitioner among different plans, all these different plans usually contain a manual activation mode and their name ends with manual. Sometimes the same action is also enforced by the protocol without the need for an explicit decision of the practitioner. In this case the plan representing the action mostly is identical to the manual version except that the manual activation mode in the form of a condition is missing. Below an example of two plans originating from the diabetes protocol that are except for this activation mode identical. Yet they do need to be specified separately in the Asbru formalization, which causes unnecessary complexity.

Plan Fasting-glucose-test-manual
Fasting glucose test
conditions
activate-mode: manual
plan-body type = any-order
wait-for all
ask glucose-measurement
ask fasting-glucose

plan Fasting-glucose-test
Fasting glucose test
plan-body type = any-order
wait-for all
ask glucose-measurement
ask fasting-glucose

In the example above as soon as ***Fasting-glucose-test-manual*** is called the practitioner executing the protocol explicitly has to decide whether or not to execute this plan (mostly with another plan as alternative). On the other hand as soon as ***Fasting-glucose-test*** is called, the plan is executed anyway.
These manual and automatic versions of a plan appear both in the jaundice protocol as well as in the diabetes protocol.
This extra complexity could be resolved by determining on a higher level (in a superplan) whether a plan is manual or automatic. In this way the actual plan on the lower level has to be defined only once and so a lot of space is saved.

5.8 Asbru control structures

There are certain Asbru control structures, required for the functionality of Asbru, that mostly are not stated in the original protocol, but have to be modeled explicitly in the Asbru protocol. They can be either derived from the original protocol or obtained from domain experts. This means these specific control structures cause extra complexity in the Asbru translation.
The plan body type is an often appearing example of such a control structure. It defines the order in which the elements of the plan body should be performed, which can be sequentially, unordered, any order, etc. Mostly this follows implicitly from the original protocol, but in Asbru an explicit statement is needed. Another related example is the subplan type, which determines the order in which the elements of the subplan body should be performed.
A final example is given by wait-for-optional-subplans, which indicates whether or not the plan should wait for other subplans to complete as long as the accompanying time annotation allows it.

5.9 User-performed plan body

A plan body can be user-performed, which means this plan is executed through some action by the user, which is not further modeled in the system. Mostly it is apparent which actions should be executed by the user so this is not explicitly stated in the original protocol. This means a user-performed plan body has no relation with the original protocol and so it can be considered as another complexity factor. The linked protocols also show that in both the jaundice and diabetes protocol the numerous user-performed plan bodies remain unlinked. For example in the jaundice-plan ***Prescribe-exchange-transfusion***:

plan Prescribe-exchange-transfusion
Prescribe exchange transfusion
plan-body user-performed

5.10 Nearly identical plans

Another example of increased complexity is caused by plans that are nearly identical except for some specific instantiations. Within the framework of global data use the situation can arise that nearly identical plans exist next to each other.
For example the three plans below originating from the diabetes protocol are almost identical except for the fact that one considers the evening dose, another the morning dose and the third both the morning and evening dose, but the procedure and the effect are the same.

```
Plan Test-glucose-and-adapt-evening-insulin
    arguments
      evening-dose: amount
    plan-body type = sequentially
    wait-for all
      Four-points-day-curve
      ask evening-adaptation
    returns (+ evening-dose evening-adaptation)
```

```
plan Test-glucose-and-adapt-morning-insulin
    arguments
      morning-dose: amount
    plan-body type = sequentially
    wait-for all
      Four-points-day-curve
      ask morning-adaptation
      returns (+ morning-dose morning-adaptation)
```

```
plan Test-glucose-and-adapt-insulin
    arguments
      evening-dose: amount
       morning-dose: amount
    plan-body type = sequentially
    wait-for all
      Four-points-day-curve
      ask evening-adaptation
      ask morning-adaptation
    returns (+ evening-dose evening-adaptation),
    (+ morning-dose morning-adaptation)
```

An alternative solution for this example, which would reduce complexity, would be one plan containing the doses as arguments and the adapted doses as a return values. Yet it always remains a decision of the modeler whether to choose for decreased complexity in stead of readability or the other way around.
The manually vs. automatically activated plans described in the previous paragraph can be considered as a form of nearly identical plans as well, since they only differ in their activation mode. Yet the difference with the plans in the example above is that in case of the manual activation there is a real difference in execution, while in case of the nearly identical plans the execution is the same, only the value on which the execution is performed differs.
It is interesting to note that these nearly identical plans have only been observed in the diabetes protocol and not in the jaundice protocol.

5.11 Domain specifications

Every Asbru protocol starts with a domain specification which describes the interface of the plan library to (parts of) the environment in which the plans will be executed. This domain specification describes for example parameters, variables and constants. The domain specification defines these different entities, but in practice they are only used later on in the protocol.

Most of the definitions in the domain specification can be related to the places in the original protocol where they are used. This mostly means they belong to a multiple link from one place in the original protocol to this definition and of course also the place where it is really used in practice. It is only the functionality of Asbru that asks for these definitions, which causes extra complexity in the form of a lot of extra pages compared to the original protocol.
Besides the diabetes protocol refers to a risk table for coronary heart diseases. It consists of 4 subtables for both men and women with or without diabetes. Depending on the patient one of the 4 tables can be used to estimate the risk of coronary heart diseases. To be able to represent these 4 tables in Asbru a lot of transformations had to be done and 5 pages of domain specifications were needed to write the representation down. Thus this risk table has caused quite a lot of extra complexity only to determine the CH-disease risk.
Tables with numbers originating from an original protocol like this risk table in diabetes are mostly represented in the domain specifications. There they often occupy a lot of space and so influence the complexity of an Asbru protocol negatively. In the future it would be a challenge to find a less complex Asbru representation for tables.

5.12 Prescription of medicine

Especially the diabetes protocol covers some prescription of medicine including the initialization and the adaptation (either increase or decrease) of a dose and also the transition of one medicine to another.
It was already suspected that the functionality of Asbru is not really suitable for the prescription of medicine. The linking task indeed showed that the plans considering the prescription of medicine are just slightly related to the original protocol and so introduce a lot of extra complexity compared to the original protocol.
Below an example from the diabetes protocol which is already 18 lines long and still only considers the initialization of a drug dose; not to mention the adaptation of a dose.

```
plan Initialize-drug-doses
  arguments
        drugs: list of String
        iterator-drugs: iterator
        drugs-doses: list of amount
        iterator-drug-doses: iterator plan
  plan-body type = sequentially
  wait-for all
        reset-iterator(iterator-drugs)
        reset-iterator(iterator-drugs-doses)
        go-to-next(iterator-drug-doses)
        do-repeatedly
           ask drug dose
           insert-before-iterator(drug-dose, iterator-drug
           doses)
           go-to-next(iterator-drugs)
           go-to-next(iterator-drug-doses)
        termination-condition: is-at-end (iterator-drugs)
  returns drugs-doses
```

In these cases of medicine prescription mostly the only links that have been defined are from the plan name to the place in the original protocol where for example the adaptation of the dose is mentioned, but further the plan remains unlinked.
Since the jaundice protocol doesn't really consider the prescription of medicine, the only reference in this case is the diabetes protocol. Though it is expected that other protocols containing medicine prescription will also need complex constructions to be able to represent it in Asbru. So it would be an interesting challenge to find a more useful and less complex representation for the prescription of medicine in Asbru.

5.13 Complexity overview

To give the reader an impression of how the different complexity aspects discussed in this chapter appear in the two Protocure protocols, the table below enumerates the appearance of every complexity issue that could be counted for both jaundice and diabetes.

Complexity issue	**Jaundice**	**Diabetes**
Conditions	14	24
Intentions	18	17
Retry delays of cyclical plans	1	5
Conversion of parameters to variables	1	3
Manually vs. automatically activated plans	2	1
Asbru control structures	28	50
User-performed plans	19	20
Nearly identical plans	-	2
Prescription of medicine	-	12

It can be seen that some of the issues appear often and some little in both protocols and some only appear in one of the two protocols. Because, as already explained in paragraph 2.3, the two protocols differ on a lot of dimensions and the diabetes protocol also is 16 pages longer than the jaundice protocol, no further conclusions about the mutual relationship between the numbers can be drawn.

6 Patterns in Asbru

Patterns are a useful technique for the transmission of knowledge about recurrent problems. A pattern is a named, reusable solution for a recurrent problem in a particular context. Patterns are not miraculous recipes that will work in every scenario, but they do convey important knowledge, a standard solution and a common language about a recurrent problem. Sometimes patterns can be identified within other patterns, so patterns can be thought of as consisting of smaller subpatterns.
Also in the Asbru protocols patterns can be identified. A pattern in Asbru is a collection of statements that together model a certain kind of situation. Once these patterns have been defined they can be reused during the translation of medical protocols in Asbru. There is a growing interest in speeding up the process of translating protocols in Asbru and the use of patterns can be a challenging way to establish this.
Patterns that appear in Asbru protocols are depending on the characteristics of the protocol and the style and preferences of the modeler. Related protocols (developed by the same institution or treating the same syndrome) or protocols modeled by the same knowledge engineer will show more corresponding patterns than non-related protocols or protocols modeled by different knowledge engineers.
At the moment already macro files modeling basic Asbru elements are developed for the Guideline Markup Tool. For example the Asbru functionality if-then-else is a pattern that has been defined as a macro. These macros for basic Asbru elements already cover a lot of the patterns appearing in the protocols.
More interesting than these basic Asbru elements is to identify Asbru patterns at a higher level, for example at plan or plan-body level. A plan structure that has been identified as a pattern can then be reused during the modeling process in the future without having to define it again every time. For the identification of patterns in medical protocols, there are two possibilities:

- patterns can be identified in Asbru protocols
- patterns can be identified in the original medical protocols

Within this project only the first possibility has been investigated. So the two reference Asbru protocols have been taken and examined for reappearing Asbru patterns. In practice it turned out the two protocols form a quite limited resource for identifying patterns and a lot of patterns are already covered by the basic Asbru elements. Besides the Protocure protocols have especially been selected, because they differ on several dimensions, which makes it harder to find corresponding patterns. So in the future it would be interesting to have more, preferably corresponding, Asbru protocols for the identification of patterns.
Nevertheless some patterns have been identified in the jaundice and diabetes protocol. They will be discussed one by one in the following paragraphs. The patterns show the general plan structure and the italic text between brackets has to be instantiated for the specific plan. On the right of each pattern also an example from one of the protocols is given.
Although it has not been investigated now the second possibility of identifying patterns in original medical protocols could be a very interesting topic for future research. It takes a different point of view as starting point. While for the first option Asbru protocols are taken as a starting point in this case the original protocols do. First reappearing patterns in the original protocols can be identified and then a universal Asbru translation for such a pattern can be devised and included in tools like the Guideline Markup Tool. Once a protocol containing this pattern has to be translated in Asbru such patterns can very easily be reused. Of course the more patterns identified and available for reuse, the bigger the speed up of the modeling.

6.1 Manually activated, alternative plans

As described earlier in chapter 5 about protocol complexity sometimes the medical practitioner has to make a manual selection, which of some plans to perform. In this case these plans should not be executed automatically once they are called, but the practitioner should explicitly decide to execute a plan or not. In Asbru this can be regulated by conditioning the activate mode of a plan. If the activate mode is automatic, the plan is started (i.e. it becomes activated) immediately after becoming ready. If the activate mode is set to manual however, the user is asked for approval before the plan is started. If the user does not answer in the time allotted by a time-out then the plan is rejected.
This selection among different plans appears more often in both the jaundice (three times) and diabetes protocol (four times). What happens is that the plan, where the selection between alternative plans has to be made (for example in the diabetes plan ***Choose-feeding-alternative***) contains the following plan body structure:

plan-body type = any-order **wait-for** one *[alternative-plan-name-1]* | *[alternative-plan-name-n]*	**plan-body** type = any-order **wait-for** one Breastfeeding Breastfeeding-with-formula Formula only

This means this plan completes as soon as one of the alternative plans has been performed, where the order in which the plans are tried is of no importance.
All the alternative plans the practitioner can choose from, like the ***Breastfeeding*** plan in jaundice, satisfy the following structure, which can be identified as a pattern:

plan *[plan-name]* *[comment]* **conditions** **activate-mode**: manual **plan-body**: *[plan body]*	**plan** Breastfeeding *Breastfeeding* **conditions** **activate-mode**: manual **plan-body**: user-performed

By combining this pattern with the plan body structure described above the medical practitioner has to decide to perform one of the alternative plans before the execution of the protocol can continue.
In the two Protocure protocols the plan body of the pattern above is mostly user-performed. There are two exceptions in the diabetes protocol where the plan body consists of one or more asks for a certain parameter. So next to the user-performed plan body, this is another instance of the pattern. The described pattern appears in both of the two protocols only in the explained context, always in combination with a superplan containing the plan body type described above.

6.2 User-performed plans

Clinical protocols often refer to clinical actions that have to be performed by a medical practitioner. In Asbru this is modeled with the following pattern (with an example about feet examination from the diabetes protocol):

plan *[plan-name]* *[comment]* **plan-body** user-performed	**plan** Feet-examination *Feet examination* **plan-body** user-performed

In case a plan like this is used, the real execution of the plan is left to the medical practitioner, so the details are out of the scope of the protocol.
This pattern often appears in both the jaundice (9 times) as well as the diabetes protocol (15 times). Though in the jaundice protocol the following specific instance of this pattern appears:

plan Exit-*[description]* *[comment]* **plan-body** user-performed	**plan** Exit-possibility-of-cholestatic-jaundice *Exit this protocol. Perform appropriate laboratory assessment (possibility of hemolytic disease)* **plan-body** user-performed

This exit construction is used when the protocol does not apply to the symptoms of a patient, but other medical measures should be taken. For example in case of the given example on the right the symptoms possibly indicate cholestatic disease, so the protocol should be abandoned to perform appropriate laboratory assessment to examine the possibility of cholestatic disease. These kind of exits do not appear as such in the diabetes protocol. Although this protocol does know instances of the user-performed pattern like ***Chiropodist-referral***, where a certain clinical action is contracted out to another medical specialist. Yet in these cases the actions performed by the other specialist do belong to the treatment while in case of the exit plans the medical actions fall outside the scope of the protocol.

6.3 Intention to know a variable

A nice feature of Asbru is that also the aim of a plan can be defined using the Asbru component "intention". Examining the two Asbru protocols shows that a lot of plans are intended to uncover the value of a certain variable, for example the diabetes-plan ***Anamnesis*** is intended to uncover if glucose determination is needed. Though this happens more often in the jaundice protocol (8 times) than in the diabetes protocol (3 times). In Asbru protocols the intention to uncover a certain variable is modeled using the following pattern:

intentions **achieve overall-state**: is-known-variable(*x*)	**intentions** **achieve overall-state**: is-known variable(glucose-determination-needed)

For a variable to be known it has to obtain a value somewhere in the plan. Sometimes this happens in a subplan, but often a variable gets its value in the plan itself with an explicit statement. Frequently the value of a variable depends on certain conditions expressed in an if-then-else statement. That is why the intention pattern above often appears in combination with the following pattern (with an example from the diabetes-plan ***Anamnesis***):

if *[condition]* **then** *[x]* ← true **else** *[x]* ← false	**if** (**or** (typical-signs = true) (risk-factor = true)) **then** glucose-determination-needed ← true **else** glucose-determination-needed ← false

6.4 Intention to know a parameter

Related to the pattern described in the previous paragraph some plans intent to uncover a certain parameter value, which is reflected by the pattern below (with an example from the diabetes-plan ***Diagnostics***):

intentions **achieve overall-state**: is-known-parameter(*x*)	**intentions** **achieve overall-state**: is-known-parameter (glucose-evaluation)

The only difference with the pattern to uncover a variable is that in case of parameters no explicit definition of their values is needed, like the if-then-else statement for variables. In general the value of the parameter to be known is determined in one of the subplans or with an ask statement in the plan itself.

6.5 Sequence of parameter requests

For diagnosis and treatment often a lot of values representing the condition of a patient are required. In Asbru these values can be obtained by using the ask functionality, which returns the parameter value asked for.
What often happens is that several parameters are necessary. In Asbru this is translated according to the following pattern (which is a subpattern of a plan):

plan-body type = any-order **wait-for** all **ask** *[parameter]* ¦ **ask** *[parameter]*	**plan-body** type = any-order **wait-for** all **ask** thirst **ask** polyuria **ask** weight-loss

So the plan body exists of a list of requests for parameters that can be executed in any order though it should wait till all the parameters are known.
This pattern often appears in both the jaundice (5 times) as well as the diabetes protocol (8 times). The example from the diabetes protocol at the right shows an instance of this pattern. It is only part of the plan ***Anamnesis-typical-signs*** that later in the plan uses the parameter values to determine if one has to do with typical signs or not.
Another pattern very similar to the one described here appears twice in the diabetes protocol and nowhere in the jaundice protocol:

plan *[plan-name]* *[comment]* **plan-body** type = sequentially **wait-for** all **ask** *[parameter]* ¦ **ask** *[parameter]*	**plan** Cholesterol-tests *Cholesterol tests* **plan-body** type = sequentially **wait-for** all **ask** total-cholesterol **ask** HDL-cholesterol **ask** tryglycerids

Again a list of parameter requests, but this time the plan only exists of these requests and now they should be performed sequentially instead of any order.
It is not clear though whether this requirement of sequential execution is really necessary in these cases. In the given example of ***Cholesterol-tests*** for example one could wonder whether it is

really crucial that these tests are done exactly in this order, perhaps any-order would also do fine. Especially for a knowledge engineer modeling a protocol it is difficult to determine whether an order is crucial or not (for safety one could always choose the sequential order, but this could also cause a decrease in efficiency). He or she will always have to rely on advice from medical experts.

6.6 Specific diabetes patterns

On one hand general patterns appearing in all protocols can be identified. On the other hand also patterns appearing only in one specific protocol can be identified. Besides the general patterns described in the previous paragraphs, some patterns specific for the diabetes protocol have been identified and will be described below.
In case of such specific patterns it would be very useful if they can be defined at the beginning of the formalization for example in the Guideline Markup Tool, to be able to use them during the translation process of the specific protocol.

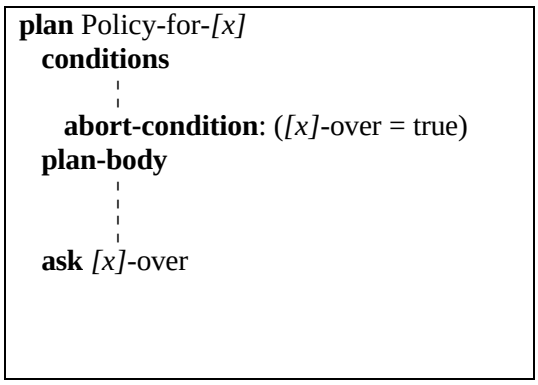
Like already described in chapter 5 about the complexity of Asbru protocols, the diabetes protocol also includes the prescription of medicine, which asks for a lot of plans controlling the drug dose. Within this framework of medicine prescription the following pattern appears:

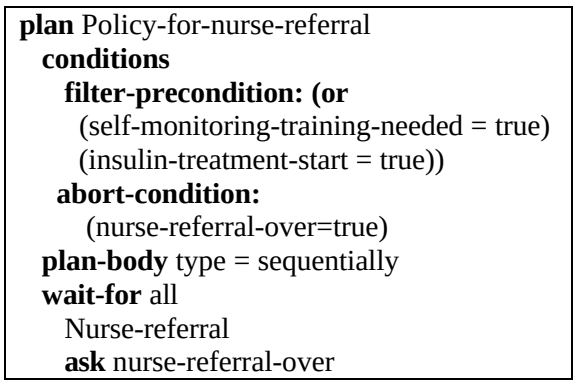
variables
[drug-name]: list of String
[drug-name]-doses: list of amount
iterator-*[drug-name]*: iterator (*[drug-name]*, first-element)
iterator-*[drug-name]*-doses: iterator (*[drug-name]*-doses, first-element)
|
|
plan-body
[drug-name] ← Initialise-drug-dose (*[drug-name]*, iterator-*[drug-name]*, *[drug-name]*-doses, iterator-*[drug-name]*-doses)
|
[drug-name] ← Find-*[drug-name]*-doses(*[drug-name]*, iterator-*[drug-name]*, *[drug-name]*-doses, iterator-*[drug-name]*-doses)

variables
antidiabetics: list of String
antidiabetics-doses: list of amount
iterator-antidiabetics: iterator (antidiabetics, first-element)
iterator-antidiabetics-doses: iterator (antidiabetics-doses, first-element)
|
|
plan-body
antidiabetics ← Initialise-drug-dose (antidiabetics, iterator-antidiabetics, antidiabetics-doses, iterator-antidiabetics-doses)
|
antidiabetics ← Find-antidiabetic-doses (antidiabetics, iterator-antidiabetics, antidiabetics-doses, iterator-antidiabetics-doses)

Within this pattern variables needed later on are defined at the beginning of the plan. Further down in the plan body first the dose is initialized and then the right dose is determined.

The diabetes protocol also contains some policies for diabetes patients in specific cases. The patient can for example be referred to another specialist. In this context the following pattern can be identified:





Within this pattern the policy plan should be aborted once for example the referral has finished. To determine when this is the case a specific parameter is used and asked for. Once this parameter becomes true, the abort condition is fulfilled and the plan will abort.

6.7 Pattern overview

Summarizing the patterns described above have been recognized in the two protocols according to the following numbers:

Pattern	**Jaundice**	**Diabetes**
Manually activated plans	3	4
User-performed plans	9	15
Sequence of parameter requests	5	8
Intention to know a variable	10	3
Intention to know a parameter	4	4
Specific diabetes patterns		
• drug regulation	-	3
• policies for diabetes patients	-	5

To be identified a pattern has to appear at least a few times in the protocols. It can be seen that the patterns described above indeed do appear more often in the protocols (at least 3 times). Only the last patterns are really specific for the diabetes protocol.
Like for the complexity issues, because the protocols differ on a lot of dimensions and the diabetes protocol is 16 pages longer than the jaundice protocol, again no conclusions can be drawn about mutual relationships between the numbers.

7 Summary

Because medical protocols can promote high quality practice, reduce variations in care and improve cost efficiency they are becoming more and more important in the medical field. However these benefits can only be achieved if the protocols are of high quality.
Within the Protocure project it has been shown the quality of medical protocols can be improved by formal methods, since a lot of anomalies were detected when some medical protocols were translated into the representation language Asbru.
This quality improvement sounds promising, but formalization turns out to be a time-consuming task, resulting in a formal protocol that is much more complex than its original version and the relation with this original protocol remains unclear. Furthermore within this graduation project it has been shown the formalization process can not only uncover anomalies in the original protocol, but can also introduce new anomalies to the formal protocol. In the last case the formalization doesn't improve the protocol, but instead makes it worse.
So quality improvement by formal methods is only promising if it can be assured the formalization represents the original protocol in the right way. Furthermore the relation between the original protocols and their formalizations, the complexity of formal protocols and time-saving measures for the formalization process, should be investigated to enable improvements.
Within this graduation project first of all the Guideline Markup Tool, developed to support the formalization process, was used to define all the links between the medical protocols selected within the Protocure project and their Asbru translations. The linking task asked for a thorough understanding of and a critical attitude to the protocols. Besides decisions had to be made about whether and how to link something.
Once the links are defined they can serve a lot of purposes. First of all they can be used to check if everything in the original protocol has really been modeled in Asbru. Second of all they can form a bridge between the medical expert and the knowledge engineer. By using the links the knowledge engineer can see which part of the formal protocol the medical expert is talking about. Third of all when an original protocol is updated the links can be very helpful in adjusting a formal protocol to changes. But most important the links provide insight in how the original protocols and their formalizations are related.
The links show the parts of the original and formal protocol that are related, but have also revealed already uncovered anomalies, new anomalies (sometimes even introduced during the formalization process), parts of the original protocol that show no relation with the formalization and finally parts of the formal protocol that show no relation with the original protocol.
Especially the fact that new anomalies have been uncovered, which had been introduced during the formalization process, is really surprising. The aim of formalization is to improve the quality of medical protocols, but apparently it can also worsen their quality, a very contradictory outcome. So like formalization can be a means to improve the quality of medical protocols, the linking can be a means to improve the quality of formal protocols.
The parts of the formal protocol that show no relation with the original protocol can be an indication for the causes of the increased complexity of the formal protocols. Within this graduation project several aspects that increase the complexity of formal protocols have been identified. Some of them appear often and others only scarcely. Some are enforced by the functionality of Asbru and others are based on a choice of the modeler. Some issues could be resolved in the future either by adjusting the representation language Asbru or by different modeling choices and others probably can't be solved.
Since the Guideline Markup Tool had never been used before it has been evaluated for its usability especially during the linking task. At first some adjustments and extensions to the tool had to be made to be able to use it for the linking at all. At the moment the Guideline Markup Tool contains the basic functionality to do the linking of the original (HTML) protocols and their

Asbru translations. Some functionality for the linking is still missing which made less elegant solutions necessary and some functionality would still be a desirable. If the tool is developed further in the future it can become a very advanced tool in assisting the Asbru formalization of medical protocols and the linking task.
Finally the interest in time-saving measures for the formalization process gave rise to the identification of reappearing patterns in the Asbru protocols. Once patterns have been identified they can be reused in the future, for example in a macro file in the Guideline Markup Tool and so speed up the formalization process. The two protocols used in the Protocure project turned out to be a scarce resource to identify patterns from. Though a few patterns have been identified and most of these patterns appear more often in both protocols.
Summarizing this graduation project has defined the relationship between the original and formal Protocure protocols and has provided a lot of new insights about this relationship.

8 Future work

Of course there is still a lot to be explored considering the formalization of medical protocols. So many pages could be written about things that can be done in the future. That is why this chapter will only focus on possible future work ensuing from the work done for this thesis. The following things could be elaborated further in the future:

- **Development of the Guideline Markup Tool**: the Guideline Markup Tool should be developed further. As described in chapter 4 there is a list of todo items of things to be implemented that will particularly improve user convenience. Furthermore some recommendations for necessary and desirable functionality of the linking task have been proposed in this chapter and will hopefully be implemented in the future.
- **Formalization of protocols using GMT**: once the Guideline Markup Tool has been developed further it would be an interesting challenge to translate a medical protocol in Asbru using the Guideline Markup Tool and all its functionalities. So not only the macros for basic Asbru elements, but also the patterns identified in this thesis and of course the linking functionality. This means links can be defined during the formalization process, which forces the modeler to think about the consequences his modeling choices on the relation between the original protocol and its Asbru translation.
- **Resolving complexity issues**: in chapter 5 a lot of complexity issues, that have been identified as a result of the linking of the protocols, have been discussed. Some of them are enforced by the functionality of Asbru and others are introduced by a choice of the modeler. For some of the issues already less complex alternatives have been proposed, for the others also less complex alternatives should be considered.
- **Identifying more Asbru patterns**: as already mentioned in the chapter about Asbru patterns, two reference protocols are a quite limited source to extract patterns from. In the future when more protocols will have been translated to Asbru, it will be interesting to see if more patterns can be identified. These patterns can then be included in a macro file for the Guideline Markup Tool and so be reused in the future.
- **Identifying patterns in medical protocols**: within this graduation project the Asbru protocols have been used as a starting point for pattern recognition. Another approach would be to take a selection of original medical protocols and identify patterns within them. Once these patterns have been identified, an Asbru translation can be formulated and included into a macro file for the Guideline Markup Tool, to be used for future formalizations.

References

1. S. Woolf. Practice guidelines: a new reality in medicine. *Archives of internal medicine*, **160**, 1811-1818, 1990.

2. M. Musen, J. Rohn, L. Fagan, and E. Shortliffe. Knowledge engineering for a clinical trial advice system: uncovering errors in protocol specification. *Bulletin du cancer*, **74**, 291-296, 1987

3. I. Graham, L. Calder, P.Hébert, A. Carter and J. Tetroe. A comparison of clinical practice guideline appraisal instruments. *International Journal of Technology Assessment in Health Care*, **16**(4), 1024-1038, 2000.

4. The Protocure project – Improving medical protocols by formal methods. http://www.protocure.org

5. M. Peleg, S. Tu, J. Blury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A, Kumar, S. Miksch, S. Quagline, A. Seyfang, E. Shortliffe and M. Stefanelli. Comparing models of decision and action for guideline-based decision support: a case-study approach (part 1 of 2), (to be published in 2002).

6. A. Seyfang, R. Kosara and S. Miksch. Asbru's reference manual, Asbru version 7.2. Technical report, Vienna University of Technology, Institute of Software Technology, 2000.

7. AAP: American Academy of Pediatrics. Provisional Committee for Quality Improvement and Subcommittee on Hyperbilirubinemia. Practice parameter: management of hyperbilirubinemia in the healthy term newborn. *Pediatrics*, 94, 558-565, 1994.

8. G.E.H.M. Rutten, S. Verhoeven, R.J. Heine, W.J.C. de Grauw, P.V.M. Cromme, K. Reenders, E. van Ballegooie and T. Wiersma. NHG-standaard Diabetes Mellitus Type 2 (eerste herziening). *Huisarts en Wetenschap*, **42**(2): 67-84, 1999.

9. M. Marcos, G. Berger, F. van Harmelen, A. ten Teije, H. Roomans and S. Miksch. Using critiquing for improving medical protocols: harder than it seems. In *Proceedings of the 8th European Conference on Artificial Intelligence in Medicine (AIME-2001)*, 431-441, Springer, Berlin, 2001.

10. The Guideline Markup Tool - http://hildegard.asgaard.tuwien.ac.at/~peter/GMT/

11. S. Miksch. Plan management in the medical domain. *AI Communications*, **12(4)**, 209-235, 1999.

12. W.M. Tierney. Improving clinical decisions and outcomes with information: a review. *International Journal of Medical Informatics*, **62**, 1-9, 2001.

13. M.J. Field and K.N. Lohr (eds.). *Clinical practice guidelines: directions for a new program*. Institute of medicine. Washington DC, National Academy Press, 1990.

14. S.W. Tu, M.G. Kahn, M.A. Musen, J.C. Ferguson, E.H. Shortliffe and L.M. Fagan. Episodic skeletal-plan refinement on temporal data. *Communications of the ACM*, **32**, 1439-1455, 1989.

15. P.E. Friedland and Y. Iwasaki. The concept and implementation of skeletal plans. *Journal of automated reasoning*, **1**(2), 161-208, 1985.

16. G. Feder, M. Eccles, R. Grol, C. Griffith and J. Grimshaw. Using clinical guidelines. *British Medical Journal*, **318**, 728-730, 1999.

17. S. Woolf, R. Grol, A. Hutchinson, M. Eccles, J. Grimshaw. Potential benefits, limitations and harms of clinical guidelines. *British Medical Journal*, **318**, 527-530, 1999.

18. M. Marcos, M. Balser, A. ten Teije, F. van Harmelen. From informal knowledge to formal logic: a realistic case study in medical protocols. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, 2002.

19. R. Kosara, S. Miksch, A. Seyfang and P. Votruba. Tools for acquiring clinical guidelines in Asbru. In: *Proceedings of 6th World Conference on Integrated Design and Process Technology (IDPT'02)*, 2002.

20. S. Miksch, Y. Shahar and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In: *Proceedings of the 7th Workshop on Knowledge Engineering: Methods and Languages (KEML-97)*, Motta, E., van Harmelen, F., Pierret-Golbreich, C., Filby, I. And Wijngaards, N., eds, Milton Keynes, UK, 1997.

21. M. Balser, W. Reif, G. Schellhorn, K. Stenzel and A. Thums. Formal system development with KIV. In T. Maibaum, editor, *Fundamental Approaches to Software Engineering*, nr. 1783 in LNCS, Springer, 2000.

22. M. Marcos, H. Roomans, A. ten Teije and F. van Harmelen. Improving medical protocols through formalization: a case study. *6th International Conference on Integrated Design and Process Technology (IDPT-02)*, 2002.

23. The Asgaard project – designing task-specific problem-solving methods to support the design and execution of time-oriented skeletal plans http://hildegard.asgaard.tuwien.ac.at

24. Y. Shahar, S. Miksch and P. Johnson. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, **14**, 29-51, 1998.

25. S. Miksch, R. Kosara, Y. Shahar and P. Johnson. AsbruView: visualization of time-oriented skeletal plans. In: *Proceedings of the 4th International Conference on Artificial Intelligence Planning Systems (AIPS-98)*, Carnegie-Mellon University, Pittsburgh, PA, AAAI Press, 11-18, 1998.

26. R. Kosara and S. Miksch. A User interface for executing Asbru plans. *Artificial Intelligence in Medicine*, 136-139, 2001.

27. T. Bosse. *An interpreter for clinical guidelines in Asbru*. Master Thesis, Vrije Universiteit Amsterdam, 2001.

28. K. Hammermüller and S. Miksch. A workflow model for the Asgaard project: a time-oriented, skeletal planning workbench in medicine, in workshop *"Computers in Anaesthesia and Intensive Care: Knowledge-Based Information Management"*, in conjunction with the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99), Aalborg, Denmark, June 20, 1999.

29. H. Roomans. *Formalization of a medical guideline. Usability investigation of the Asbru modeling language*. Master's thesis, Universiteit van Amsterdam, 2001.