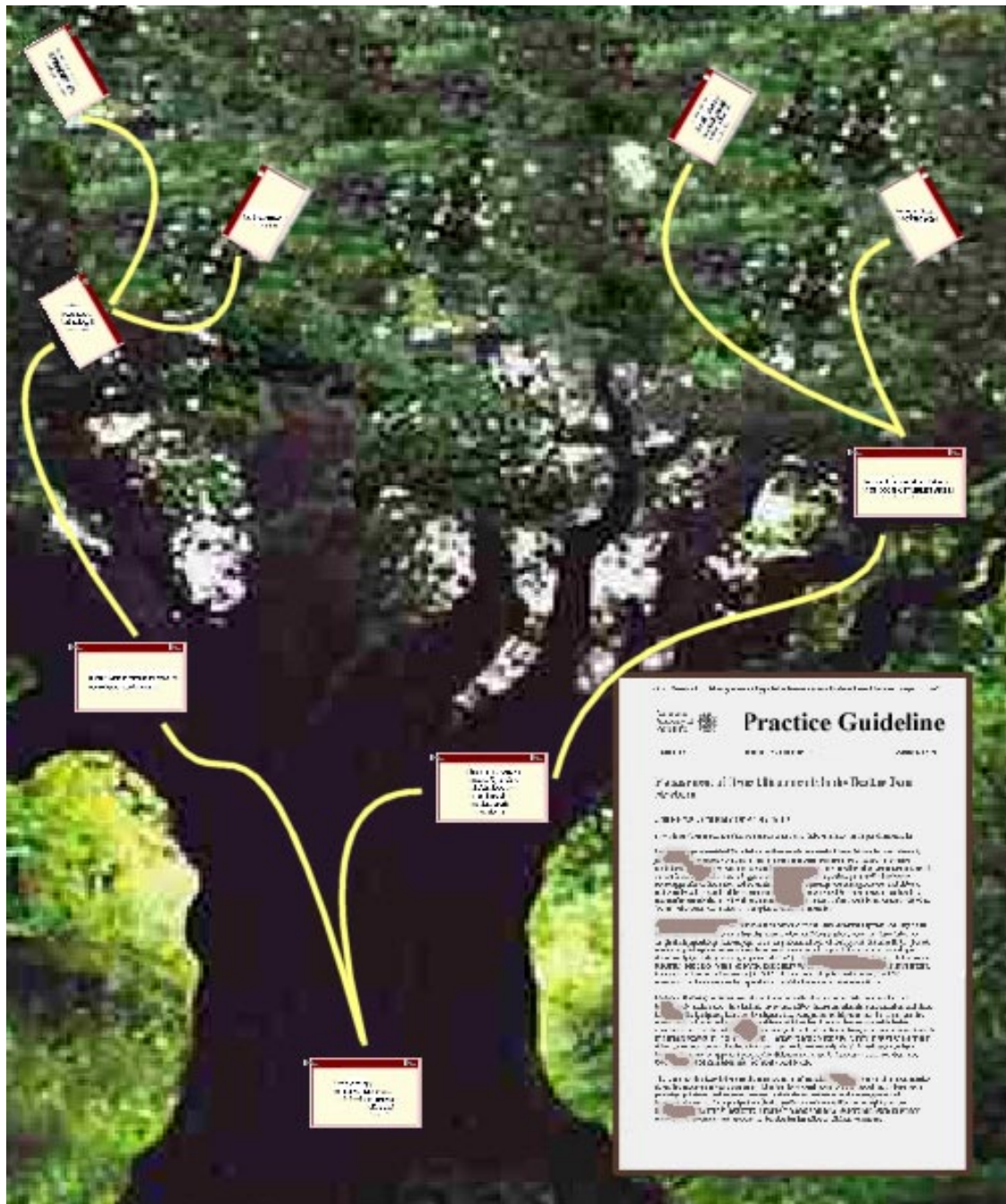


Formalisation of a medical guideline

Usability investigation of the ASBRU modeling language



Formalisation of a medical guideline

Usability investigation of the ASBRU modeling language

Student

H.F. Roomans
Van Gentstraat 10-3
1055 PE Amsterdam
Collegekaart nummer: 9556443
E-mail: hugo@roomans.net

Mentors

Dr. A.C.M. ten Teije
Informatica Instituut
Vakgroep Intelligente systemen

Dr. F. van Harmelen
Divisie der Wiskunde en Informatica ..
Afdeling Kunstmatige Intelligentie VU

Traineeship address

Vrije Universiteit
Faculteit der exacte wetenschappen
Divisie W&I
De Boelelaan 1081-83
1081 HV Amsterdam

Supervisor

Dr. A. Abu-Hanna
Faculteit Geneeskunde
Vakgroep Klinische Informatiekunde AMC-UvA

Practice teaching period

October 1999 – January 2001

Samenvatting

Het belang van medische richtlijnen wordt steeds groter, en zo ook hun aantal. De hoeveelheid richtlijnen en hun complexiteit hebben van gecomputeriseerde ondersteuning een belangrijk onderwerp gemaakt. Door gebruik te maken van computers kunnen niet alleen aspecten van richtlijnmanagement worden ondersteund, ook nieuwe toepassingen van richtlijnen worden mogelijk. Voorbeelden van zulke nieuwe toepassingen zijn richtlijn gebaseerde beslissingsondersteuning, en critiquing systemen voor het bepalen van conformiteit tussen klinische praktijk en richtlijn aanbevelingen of intenties. Op het moment zijn er een aantal voorgestelde benaderingen met als doel medische richtlijnen te formaliseren om hun gecomputeriseerde gebruik te verbeteren. Eén van deze benaderingen is Asbru, het onderwerp van onderzoek in deze studie. Asbru kent meerdere maten van formalisatie, van het structureren van de richtlijn in componenten, tot meer vergaande specificaties in bijvoorbeeld temporele patronen. Asbru kan worden gekarakteriseerd door de nadruk op planning taken, temporele aspecten, en een expliciete representatie van intenties.

Het doel van deze studie is tweevoudig. Ten eerste wordt de bruikbaarheid van Asbru onderzocht ten aanzien van het modeleren van medische richtlijnen. Ten tweede is onderzocht of en in welke mate het proces van formaliseren helpt bij het opsporen van onvolkomenheden in een richtlijn. Het onderzoek is uitgevoerd aan de hand van het formaliseren van een specifieke richtlijn als een case studie.

De medische richtlijn die gemodelleerd is met behulp van de Asbru taal, is de richtlijn over hyperbilirubinemie van de American Academy of Pediatrics. Deze richtlijn, die is opgenomen in de National Guideline Clearinghouse, kan worden beschouwd als representatief voor het merendeel van de richtlijnen. Het modelleringsproces heeft geresulteerd in een model van de richtlijn, en inzicht in Asbru's sterke en zwakke eigenschappen. Dit heeft geleid tot een aantal aanbevelingen met betrekking tot het ontwerp van Asbru. Asbru's bruikbaarheid wordt als redelijk beoordeeld, hoewel een aantal aspecten als gebruiksonvriendelijk gezien worden. Asbru heeft echter voldoende mogelijkheden om de bruikbaarheid naar een hoog niveau te tillen. Het introduceren in Asbru van de in deze studie gedane aanbevelingen zal zeker bijdragen aan een verbeterde bruikbaarheid. Op het moment van dit schrijven bevat de meest recente versie van Asbru al een aantal van de in deze studie gedane aanbevelingen.

Het tweede onderzoek, naar de rol van het formaliseer proces in het inspecteren van de interne kwaliteit van richtlijnen, bracht een significant aantal onvolkomenheden van de richtlijn aan het licht. Er werden drie soorten onvolkomenheden opgespoord: 'meervoudig interpreteerbare richtlijn delen', 'ontbrekende informatie', en 'tegenstrijdigheden'. Aangezien de onderzochte richtlijn afkomstig is van een team van professionals en is opgenomen in het archief van de National Guideline Clearinghouse, kan worden aangenomen dat deze onvolkomenheden niet eenvoudig kunnen worden opgespoord zonder formalisatie.

De hier gepresenteerde studie, die heeft geleid tot één van de eerste complete richtlijn modellen in Asbru, heeft bijgedragen aan het begrip ten aanzien van de bruikbaarheid van Asbru als taal om medische richtlijnen mee te formaliseren. De aanbevelingen die gebaseerd zijn op deze opgedane kennis hebben bijgedragen aan het verbeteren van Asbru's bruikbaarheid op dit gebied. Het formalisatie proces bleek een adequate mogelijkheid te bieden voor de verificatie van de interne kwaliteit van de oorspronkelijke richtlijn. Er is veel werk te verrichten op het gebied van richtlijn formalisatie, maar deze studie en de resultaten daarvan wijzen op veelbelovende onderzoeksmogelijkheden.

Abstract

As the importance of medical guidelines is increasingly growing, so is their number. Their quantity and complexity have made computerised support a topical subject. By means of computers not only the administrative aspects of guideline management are facilitated, it also allows for new ways to use guidelines. Examples of such new uses include guideline-based decision support in e.g. treatment planning, and critiquing systems for assessing conformance of clinical practice with guideline prescriptions or intentions. Currently there are a number of suggested approaches aiming at formalising medical guidelines to enhance their computerised use. Among these approaches is Asbru which forms the object of investigation of this thesis. Asbru provides various degrees of formalisation ranging from structuring the guideline into components to more rigorous specifications of e.g. temporal patterns. It can be characterised by its focus on planning tasks, its temporal aspects, and explicit representation of intentions.

The aim of this study is twofold. The first is investigating Asbru's usability for modelling medical guidelines. The second is investigating to which extent the process of formalisation helps to spot weaknesses and irregularities within a guideline. The investigations have been carried out by means of formalising a specific guideline as a case study.

A medical guideline issued by the American Academy of Pediatrics on the subject of hyperbilirubinemia has been modelled using the Asbru language. This guideline, which has been accepted by the National Guideline Clearinghouse, can be considered representative for a majority of other guidelines. The modelling process has resulted in a guideline model, and insight into Asbru's modelling merits and weaknesses. This investigation has led us to suggest a number of recommendations on Asbru's design. Asbru's usability was found to be reasonable although it is user unfriendly in various aspects. Asbru has the ability however to increase its usability to high levels. The introduction in the language of the recommendations made in this study would definitely help to achieve this increase of usability. At the time of this writing the most recent version of Asbru incorporates a number of the recommendations made in this study.

Our second investigation to the role of the formalisation process in guideline quality inspection has indeed revealed that a significant number of guideline irregularities can be identified. In our case study three different kinds of irregularities were identified: 'ambiguous guideline parts', 'missing information' and 'inconsistencies'. As the guideline in question was issued by a team of professionals, and accepted into the archive of the National Guideline Clearinghouse it can be assumed these irregularities are not to be easily detected without formalisation.

This study, which has resulted in one of the first full guideline models to be written in Asbru, has contributed to understanding the weaknesses and merits of the usability of Asbru as a medical guideline formalisation language. The recommendations based on this understanding have contributed to its increased usability. The formalisation process was found to provide an adequate means for the verification of the internal quality of the original paper-based guideline. There is much further work to be done in the field of guideline formalisation but our investigations and results point at new promising research directions.

Acknowledgement

Besides my mentors Frank van Harmelen and Annette ten Teije, and my supervisor Ameen Abu-Hanna, a special thanks is due to the domain expert Frea Kruisinga for the necessary time investment and patiently answering all the questions, to Geert Berger for our co-operation and for showing me where to get some decent coffee ☺, to Hilco Prins for his interest in this study and for reading and commenting on all previous versions of this paper, to Mar Marcos, for her thorough examination of the model, the content of the paper and its grammar, and last but not least to Silvia Miksch, Andreas Seyfang, Robert Kosara and Georg Duftschmid from the university of Vienna for their hospitality, and their interest in this study.

Besides this scientific support, I would like to thank my girlfriend Liesbeth Salm and my parents for their personal support and believe in me.



Hugo Roomans,
Amsterdam, 30-01-2001.

Content

Subject	Page
1 Introduction	2
1.1 General introduction	2
1.2 Objective of the study	3
1.3 Structure of this thesis	3
2 Guideline representation languages	4
2.1 About Guidelines	4
2.2 GLIF	5
2.3 PROForma	8
2.4 Asbru	10
2.5 Comparison	13
3 Asbru in detail	15
3.1 Plan Body	16
3.2 Time Annotation	17
3.3 Condition	18
3.4 Intention	19
3.5 Effects	20
3.6 Preferences	21
4 Guideline formalisation	22
4.1 Material and methods	22
4.1.1 Knowledge acquisition	22
4.1.2 Modelling process	23
4.2 Results	25
4.2.1 Knowledge acquisition	25
4.2.2 Modelling process	28
4.3 Discussion	38
4.4 Conclusion on Asbru usability for guideline formalisation	40
5 Guideline verification	42
5.1 Materials and methods	42
5.2 Results	43
5.3 Discussion	45
5.4 Conclusion on guideline verification by formalisation	47
6 Reflection	48
7 Bibliography	50
8 Appendix	53
A: Guideline ' <i>Management of hyperbilirubinemia in the healthy term newborn</i> '	54
B: Guideline Model, Asbru version 7.2	65

1 Introduction

1.1 General introduction

It was already in the 4th century BC when Plato engaged in a thought experiment where doctors were “no longer allowed unchecked authority”, but had to form councils whose conclusions would “dictate the ways in which the treatment of the sick is practised” [1,2]. Although Plato was not without reservations, his experimental idea precedes in many ways what we now know as a clinical practice guideline.

It was not until the last decades of the 20th century that clinical guidelines were issued in increasing extension, and only in the past decade has their significance grown rapidly [3]. As most healthcare systems today are faced with rising healthcare costs, fuelled by an ageing population, more expensive technologies and a desire to provide the best care possible, there are multiple reasons for this increasing interest in clinical guidelines [4]. It is generally believed that the creation of and adherence to guidelines can be a tool against the delivery of inappropriate or inefficient care, and a means to improve the cost-efficiency of the delivered care. This belief has been supported by studies indicating that adherence to guidelines might reduce the costs of healthcare up to 25% [5]. Other evaluations show that clinical guidelines can indeed be a tool to improve the quality of care [6], especially as they can incorporate the latest standards of evidence based medicine. The scientific evidence supporting the benefits of guidelines, and the fact that guidelines can be used to introduce evidence based medicine into everyday medical practice have fuelled the increasing amount of guidelines even further.

The great number and complexity of guidelines, combined with the wish to increase adherence of guidelines have made automated support by use of computers a topical subject. To incorporate a guideline in a computer application, the information contained by the guideline has to be translated from natural language into a format understandable to the computer. Specifying guidelines with enough detail for computer use can be difficult because greater precision may be required than is typically found in paper descriptions [10]. As smaller details need modelling, the representation format needs greater expressiveness, though must continue to be formal enough to ensure executability. At present a number of these ‘formats’, also called ‘Knowledge Modelling Languages’ or ‘guideline representation languages’, are being developed specifically for the medical domain, such as Asbru [7], Glif [8] and PROForma [9]. Although these languages differ in their approach, all try to achieve a correct, computer based guideline representation. With this correct representation different goals can be pursued, such as the creation of a decision support system or a critiquing system. It is also believed that the formalisation process and results can help discover hiatus’ in the original paper-based guideline, thus contributing to the increase of the internal guideline quality. The increase of guideline quality, combined with a means to increase adherence of the guideline should lead to an improved deliverance of healthcare.

1.2 Objective of the study

The general objective of this study is to contribute to the development of the ASBRU modelling language. This will be done based upon the following two research questions.

Research question 1:

Investigation whether the Asbru language is usable for representing medical guidelines, both theoretically and practically”

Relevance of research question 1:

Evaluation of the current possibilities of the Asbru modelling language, and to contribute to its development.

The distinction between theory and practice stems from the fact that something theoretically possible is not automatically practically feasible. Thus the theoretical question will be: “Is there a way of representing all necessary aspects of the guideline in the Asbru language?”, as opposed to the feasibility of modelling those theoretically possible aspects.

Research question 2:

Does formalising a guideline in Asbru help verifying this guideline?

Relevance of research question 2:

Investigate in the interest of future research whether the formalisation route provides a possibility to increase the overall quality of an original paper-based guideline.

This question concerns itself with the (non-rigorous) investigation of the possibility of finding hiatus’ or other imperfections in the original guideline, due to the formalisation route.

The Asbru language has been chosen because it is a relatively new approach and constantly developing, a situation where the idea of contributing to the language is still opportune. Asbru is a formal language which allows detailed and structured modelling, as will be seen in chapter 2, making it also a good candidate in respect to research question number two.

1.3 Structure of this master thesis

The following chapter provides a short introduction to guidelines in general. In addition to the Asbru language two other approaches are presented and a comparison between the three is made. This not only provides an overview of related work, but also helps to determine the more characteristic properties of the Asbru language. Chapter 3 will then provide a much more detailed overview of the Asbru language, as it is the main subject of this study. The two research questions, presented in the previous paragraph, will be dealt with separately. Chapter 4 is dedicated to the first research question: the method(s) used, the results, the discussion and the conclusion. This is followed by chapter 5 which has a structure similar to chapter 4, but is dedicated to the second research question. Chapter 6 is a reflection on guidelines, guideline modelling languages and their properties.

2 Guideline representation languages

2.1 About guidelines

Before exploring the field of the guideline representation languages, it is important to clarify the meaning of the term 'guideline'.

Most commonly, the term guideline is used as an abbreviation of the term 'practice guideline', or the full term 'clinical practice guideline'. The following definition of a clinical practice guideline will be used throughout this paper:

"Clinical practice guidelines are systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances." [11]

The term 'protocol' is often used equivalent with 'guideline', though guidelines tend to be more declarative whereas protocols tend to be more procedural. The procedural content of protocols tends to be more detailed, which is probably the reason why some see protocols as a more detailed version of a guideline [12]. There is no consensus on the meaning differences of both terms. To avert confusion, this thesis will only use the term 'guideline' as defined above.

In 1992 The Institute of Medicine divided guidelines into 5 different types, categorised by purpose [13]. The guideline types and examples provided by the institute's report are :

- 1 *Screening and prevention.*
For Example: Vaccination for pregnant women who are planning international travel.
- 2 *Diagnosis and prediagnosis management of patients.*
For Example: Evaluation of chest pain in the emergency room.
- 3 *Indications for use of surgical procedures.*
For Example: Indications for carotid endarterectomy.
- 4 *Appropriate use of specific technologies and tests as part of clinical care.*
For Example: Use of autologous or donor blood for transfusions.
- 5 *Guidelines for care of clinical conditions.*
For Example: Management of patients following coronary-artery bypass graft.

Besides these 5 types, another way to characterise guidelines was also recognised, namely the format in which the guidelines knowledge is presented. Formats demonstrated ranged from narrative text, tables and flowcharts to graphs, maps, lists and photographs. Although most guidelines will be a composition of multiple formats, all formats must be representable by the guidelines representation language, which makes this a very interesting characterisation as well.

Just like there are different types of guidelines, there are different types of guideline representation languages. Besides Asbru, included since it is the object of this study,

two other languages have been chosen to be presented and compared in the next sections (2.2-2.5). These two languages, *PROForma* and GLIF, have been chosen because they are both among the most relevant modelling efforts so far, and differ in their approaches not only from Asbru but also from each other. *PROforma* is more procedural and GLIF is an interchange format, whereas Asbru is very declarative. Hence, the three chosen languages present quite a nice coverage of the available spectrum of guideline representation languages. The comparison will be based upon the following characteristics (adapted from [8]) of either the modelling language or aspects generally present in guidelines.

- Background and purpose of the language.
Differences in the modelling languages could originate from a difference in goals a language pursues. In turn, the background of a language can influence these goals.
- Basic structures and components for modelling (medical-)knowledge.
These largely define how the medical knowledge is represented, and what kind of knowledge can be represented.
- Representation of sequences of decisions and actions.
Guidelines are primarily a description of advised medical actions and decisions, with an ordering imposed on them. This ordering is very important to provide correct care, and therefore an important aspect of a modelling language.
- Representation of eligibility criteria and criteria for making transitions between events.
Since there is a lot of patient-variance, criteria exist for example on when a certain 'event' is considered appropriate and when it is not, or when a certain action is considered completed or not.
- Guideline modelling process.
The 'guideline modelling process' concerns itself with the availability of tools to aid the modeller, with methodological support for model construction and with the end result of the modelling exercise.

2.2 The GLIF approach

Background and purpose of the language.

GLIF stands for the GuideLine Interchange Format. It is a format for representing guidelines in a machine-readable format that has been created by researchers from the InterMed Collaboratory, a collaboration of four U.S. universities (Columbia, McGill, Harvard and Stanford). The main goal behind the development of GLIF is to create 'a standard for representing guidelines that facilitates guideline-sharing across software tools at different medical institutions that manipulate, analyse or otherwise compute with an electronic representation of a health care guideline' [8].

GLIF is based on the experience gained by analysing four existing guideline representation systems (MLM-Arden, EON, MBTA and GEODE-CM). Based on this

experience a set of requirements was formulated, which was kept in mind during the design period of GLIF. Currently InterMed is working on GLIF 2.5_beta2, but since at present only a draft version exists, this paragraph is based on the GLIF 2.0 version 3 specification.

Basic structures and components for modelling (medical-)knowledge.

The GLIF approach consists of a specific data model, the *GLIF model*, and a *GLIF syntax* that specifies the format of the text file which contains the encoding. To encode a guideline according to the GLIF model, authors express the guideline knowledge using this GLIF syntax. The syntax is based on a generic data interchange format called ODIV (Object Data Interchange Format)[15,16], allowing the representation of object instances in text. The GLIF model consists of a set of classes for guideline entities, attributes of those classes, and data-types for the attributes values. The most important object of the model is called the *GLIF guideline object*, and contains a name, a list of authors, a characterisation of the guidelines intention, a specification of the patient eligibility criteria, an unordered list of all steps in the guideline, an indication of the starting step in the guideline, and a list of supporting didactic material.

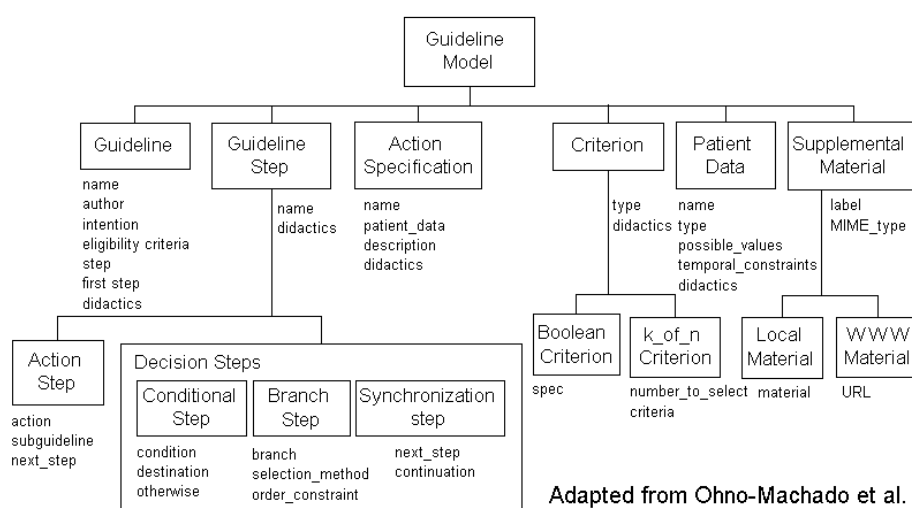


Figure 1 GLIF classes and attributes [8]

Figure 1 presents a hierarchical view of the GLIF classes and attributes. Children inherit attributes from their parent. Steps can be 'action steps' or 'decision steps', the latter being divided into three different types. As one can see in figure 1, both the action step and the synchronisation step have a pointer attribute called 'next_step', whereas the conditional step has a 'destination' attribute. The actual guideline specification will consist of a collection of steps, linked together in a directed graph. Action steps specify clinical actions that are to be performed in the patient-care process. They can contain a sub-guideline, which provides greater detail for the action. Action steps always contain exactly one action specification. If patient data is needed, a set of patient data can be associated with the corresponding action specification.

Decision steps are divided into:

- Conditional steps, directing the flow from one guideline step to another

- Branch steps, directing the flow to multiple guideline steps, and
- Synchronisation steps, converting the flow from multiple guideline steps (after a branch step) into a single step

Basic concepts of (medical-) knowledge like the intention of a guideline, the specification of an action, temporal information and criterion logic do exist, but are at present represented in free text. Noteworthy is the way didactics and supplemental material is already intertwined within the hierarchical structure of GLIF (figure 1).

Representation of sequences of decisions and actions.

Sequences of decisions and actions can be defined in GLIF by specifying a starting step of the guideline. This starting step functions as an entrance to the collection of steps, the main part of the actual guideline specification. All the steps, including the starting step, contain a pointer to their successor step(s), according to the specified sequence.

Representation of eligibility criteria and criteria for making transitions between events.

Conditional steps, directing the flow from one step to another, contain a condition, also called 'criterion'. A criterion is a logical statement that can be evaluated as true or false. True means continuation to the step defined in the 'destination attribute', false means continuation to the step defined in the 'otherwise attribute'. If there's not enough data available to evaluate the criterion, no transition occurs. In the latest full version of GLIF, the conditional expressions are represented as strings. A formal syntax for these representations is still needed, since they "*...do not have natural language processors that can parse and make sense out of such clauses*" [8]

Guideline modelling process.

A number of guidelines have been modelled with GLIF. Examples are guidelines for influenza vaccination, cholesterol screening and management, breast-mass work-up, and breast-cancer treatment. According to the five IOM guideline types mentioned in paragraph 2.1, these guidelines cover type 1 (influenza, cholesterol), type 2 (breast-mass work-up) and type 5 (cholesterol, breast-cancer treatment) [16]

There are no specific tools available to guide the modelling process, though it is fairly easy to start with a graphical representation, which is made up of the four possible steps with a small step-description (e.g. A conditional step with 'Age<10') .

The result of the modelling effort is a guideline representation in GLIF-syntax. (Figure 2) This syntax is not directly executable, but needs parsing before it is possible to "*...manipulate, analyse or otherwise compute with an electronic representation of a health care guideline*".

Guideline Example

```
{ name = "Guideline for Vaccine X";
  authors = SEQUENCE 1 {"Mary Doe, MD"};
  eligibility_criteria = NULL;
  intention = "Decide whether to recommend the Generic vaccine and at what dosage";
  steps =
    SEQUENCE 8
    { (Branch_Step 1);          (Action_Step 1);          (Action_Step 2);
      (Synchronization_Step 1);(Conditional_Step 1);      (Conditional_Step 2);
      (Action_Step 3);          (Action_Step 4);
```

```

    };
    first_step = (Branch_Step 1);
    didactics =
        SEQUENCE 1
        {
            Supplemental_Material 1
            { label = "critique";
              MIME_TYPE = "text/plain";
              Material = "Published guideline does not contain explicit eligibility criteria.";
            };
        };
}
// _____
Branch_Step 1
{ name = "collect data";
  branches =
    SEQUENCE 2
    { (Action_Step 1);
      (Action_Step 2);
    };
  selection_method = all_of;
  order_constraint = any_order;
  didactics = SEQUENCE 0 {};
}
// _____
Action_Step 1
{.....

```

Figure 2 Example of a GLIF encoding, adapted from [8]

2.3 The PROforma approach

Background and purpose of the language.

PROforma is a methodology created by the Imperial Cancer Research Fund in association with members of the ACTION cluster of projects, and has been developed within PROMPT (Protocols for Medical Procedures and Therapies). The Arden syntax, an attempt to achieve a standard format to describe medical knowledge, had a number of significant limitations [17], and the PROforma approach wanted to address these limitations, as well as some other requirements. The main goal of PROforma is to support the management of clinical procedures and clinical decision making. At present a new tool from the Infermed company called 'Arezzo' is commercially available. Arezzo is based upon the PROforma language, and is used to 'develop and maintain advanced clinical systems [18].

Basic structures and components for modelling (medical-)knowledge.

The PROforma technology consists of a graphical knowledge editor for the creation of guidelines, and an enactment engine for the execution. The actual representation of the guideline is started with the creation of a graphical high-level guideline structure. In PROforma, a guideline is specified by tasks and collections of tasks. A task can have four basic subclasses:

- Plan: A plan can contain any number of tasks, and is the basic building block of a guideline. The tasks a plan contain can be of any of the four types, including other plans. Usually a plan has some ordering imposed to reflect certain constraints, like temporal or logical constraints.
- Decision: A decision includes a set of options, a set of arguments to determine which of the options should be chosen, and 'commitment rules' to determine when the decision should be taken.
- Action: A procedure that is linked to a clinical action, for example administering an injection.
- Enquiry: An enquiry is a request for information, needed to complete a procedure or take a decision. It includes a description of the information required and a method for getting it

Each of the subclasses has a number of attributes, some are common attributes, others are specific to the sub-class. The result of this graphical modelling process is an intermediate representation, in between the informal description of the original guideline, and a machine executable knowledge base. The next step in the *PROforma* modelling process is the transcription of this intermediate model into a formal, machine readable format. This transcription is facilitated by use of templates. *PROforma* provides templates for all the specific classes of tasks, to guide the designer in formalising the medical knowledge. The guideline is only fully defined when all remaining task-templates are completed.

Representation of sequences of decisions and actions.

One of the subclasses of a task is a plan. As shown previously, a plan can contain any number of task-subclasses. It is this hierarchical structure that allows for the representation of sequences. Within a plan an ordering can be imposed. This can be done using scheduling constraints (A certain subtask can not be activated until another subtask is completed) or by use of temporal constraints. It is also possible to create cyclical sequences, using attributes of a plan-task to define the number of cycles, or to define a condition to stop the repetition. To define when the next task repetition should start, a time-interval can be given.

Representation of eligibility criteria and criteria for making transitions between events.

Among the common attributes of tasks there are three kinds of conditions:

Preconditions: to be met before a task can be started.

Postconditions: become true when a task has finished.

Trigger conditions: can start a task in spite of scheduling constraints.

Besides these three types of conditions within the common attributes, there are also two conditions within the specific attributes of plan-tasks:

Abort conditions: when met, the plan is aborted, as are all of its subtasks.

Terminate conditions: when met, the plan is finished.

Guideline modelling process

First a graphical representation is created (figure 4). This originates in a high level description of the guideline. It can be made up of the four basic building blocks of *PROforma*:

Plan: symbolised by a rounded rectangle

Decision: symbolised by a circle

Action: symbolised by rectangle

Enquiry: symbolised by a diamond.

These blocks are linked by arrows, which indicate a scheduling constraint.

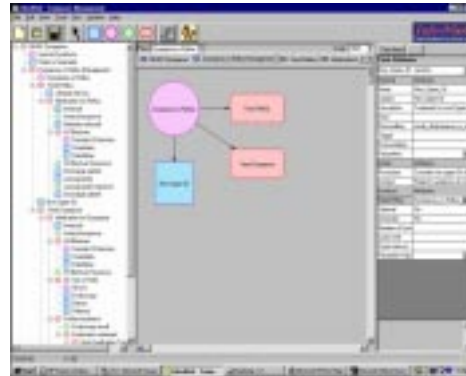


Figure 4 Screenshot of graphical representation in the Arezzo composer [18]

As mentioned in the paragraph concerning the basic structures, every sub-task inherits the same attributes from the general class 'tasks' (e.g. Title, identifier), but has also a number of specific attributes that distinguish it from the other types of tasks (e.g. Abort conditions or commitment rules). For each of these types a predefined template is available (figure 5).

This means every symbol in the graphical representation is actually a representation for a certain template. By filling in all templates the guideline representation is completed.

In *PROforma* such a completed guideline representation is immediately executable.



Figure 5 Example of a decision-task ' template [9]

2.4 The ASBRU approach

Background and purpose of the language.

The ASBRU approach is part of the Asgaard project [19], in which task-specific problem solving methods are developed, capable of performing design, execution and critiquing tasks in the medical domain. Within this Asgaard project a suitable modelling language was needed, specific to the set of plan management tasks [7]. Like the *PROforma* approach the need for this new modelling language was partly based on the experienced significant limitations of the Arden syntax [20]. ASBRU is described as 'a time oriented, intention based, sharable language, that can be used to design specific plans, as well as support the performance of different reasoning and executing tasks [7]. At the beginning of this study the most recent ASBRU version available was 6.5b, therefore this paragraph is based on the ASBRU 6.5b specification.

Basic structures and components for modelling (medical-)knowledge.

Clinical guidelines are modelled in ASBRU as a set of hierarchical skeletal plans, see chapter 3 for more details on this. These plans are the basic building blocks for representing a clinical guideline. They are identified by a unique name, and consist of the following major components:

- Time-annotations: This allows specification of a duration of time, related to a reference annotation. It can be used for two tasks: the annotation of actions, and the annotation of observing states. The time shift is specified as a temporal shift to its reference annotation. This way its possible to represent uncertainty in starting time, ending time and time duration. The reference annotation allows for multiple timelines. Time annotations are a basic structure, but not a separate component. A time annotation can be a property of an intention, condition, effect or plan-body.
- Preferences: This can constrain the selection of a plan to achieve a given goal (resources), or express a kind of behaviour of the plan (strategy). Besides resources and strategy, one can define a set of utility measures, define a look-ahead-flag to specify the number of plan-levels that have to be processed during plan-execution, and specify how precisely the conditions of a plan have to be met using the select-method.
- Intentions: The set of intentions models the aim of the plan. They are temporal patterns of actions or states, that either hold during the plan, or after finishing the plan, and can specify goals to achieve, to maintain or to avoid. This means 12 different kind of intentions are possible, all with certain time dependencies.
- Conditions: This is the control mechanism for executing plans. Like intentions they are temporal patterns, that need to hold at a certain plan step to induce a particular state transition of the plan-instance. There are seven different conditions: Filter and Setup (these are preconditions), Activate, Suspend, Reactivate, Complete and Abort conditions. There is another kind of condition that is a bit special, and in the literature it is placed separately from the other seven: the Continuation condition, since it is not based on parameter propositions. The continuation condition is elaborated on in chapter three.
- Effects: This enables to model functional relationships between plan arguments and measurable parameters, and allows the modelling of 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.
- Plan-Body: This holds the sub-plans and actions of the plan, and provides a means to represent an ordering of these sub-plans and actions.

Representation of sequences of decisions and actions.

As mentioned above, within the plan-body an ordering can be induced on sub-plans and actions. There are three types of plans: Sequential, concurrent and cyclical.

A plan-body can contain only one type of plan. The concurrent plans can be executed in parallel or any order. It's also possible to state whether or not all plans need to be completed to continue. Enumerated: the possibilities are:

DO-ALL-TOGETHER	DO-SOME-TOGETHER
DO-ALL-ANY-ORDER	DO-SOME-ANY-ORDER
DO-ALL-SEQUENTIALLY	DO-SOME-SEQUENTIALLY
CYCLICAL	

```
(PLAN I-RDS-therapy ...
(DO-ALL-SEQUENTIALLY
  (initial-phase)
  (one-of-controlled-ventilation)
  (...)))
(PPLAN one-of-controlled-ventilation ...
DO-SOME-ANY-ORDER
  (controlled-ventilation)
  (permissive-hypercapnia)
  (crisis-management)
  (CONTINUATION-CONDITION controlled ventilation))
PLAN controlled ventilation
(PREFERENCES (SELECT-METHOD BEST-FIT))
(INTENTION: INTERMEDIATE-STATE
  (MAINTAIN STATE (BG) NORMAL controlled ventilation *)) ...
(SETUP-PRECONDITIONS
  (PIP (<= 30) I-RDS *now*) (BG available I-RDS
    [_,_], [_,_], [1 MIN,_] (ACTIVATED initial-phase-1#)))
(ACTIVATED CONDITIONS AUTOMATIC) ...
(COMPLETE-CONDITIONS
  (FIO2 (<=50) controlled-ventilation [_,_], [_,_], [180 MIN,_, *self*])
  (PIP (<=23) controlled-ventilation [_,_], [_,_], [180 MIN,_, *self*]) ...
  (STATE (patient) (NOT DYSPNOEIC) controlled-ventilation
    [_,_], [_,_], [180 MIN,_, *self*]) ...
(DO-ALL-SEQUENTIALLY
  (one-of-increase-decrease-ventilation)
  (observing)))
```

Figure 6 Example of the I-RDS guideline in ASBRU, adapted from [12]

Representation of eligibility criteria and criteria for making transitions between events.

In ASBRU there are two kind of preconditions, the filter and setup conditions. Filter conditions need to hold for the plan to be applicable, but can never be achieved (e.g. patient is male). Setup conditions need to be achieved to enable a plan to start (e.g. a certain test must have been done). Criteria for making transitions between events are represented by the other conditions: when a plan is completed, when it should stop, when to pause it, reactivate it and how to start it.

Guideline modelling process.

The ASBRU 6.5b version is completely defined in Backus-Naur Form (BNF). A graphic editor called AsbruView [21] is in development. In contradistinction to the *PROforma* graphical knowledge editor AsbruView is not meant to create an intermediate model, but to design the topological and temporal views of plans written in ASBRU. The BNF-based model needs parsing before it's executable. A few clinical guidelines have been modelled, one concerning 'Infants respiratory distress syndrome' (I-RDS), and a general practitioners sinusitis guideline.

2.5 Comparison

In the previous paragraphs of this chapter, three modelling languages have been introduced. Of course they have not been minutely presented to you since this would take up significantly more pages, but it does provide enough information about the major differences between them. The differences between the GLIF and PROforma approaches and Asbru will be points of interest within research question one.

- Background and purpose of the language.

The background and purpose of all three languages are quite similar. It is clear that their main predecessor has been the Arden syntax. Its shortcomings, sometimes combined with those of other approaches, have been a starting-point for the development of all three.

Another similarity is their focus on the medical domain. It is not quite clear why these languages are specifically meant for the medical domain, and could not be used in other domains. Although differences in purpose exist, such as the GLIF desire to function as an interchange, and PROforma's wish to facilitate decision making, all want to create a correct electronic representation of a guideline in order to compute with it, making this an interesting comparison.

- Basic structures and components for modelling (medical-)knowledge.

The Asbru language places special emphasis on both intentions and temporal information. In the comparison it becomes evident that although these notions are present in all three approaches, Asbru has the most extensive possibilities. Intentions in Asbru are not limited to the intention of the entire guideline, but every plan can have its own intention. The temporal information allows for uncertainty in starting time, ending time and duration, and multiple timelines are supported based on a variety of possible reference points. To what extent these extensive possibilities are valuable remains to be seen, as they might account for a more laborious modelling. A strong aspect of the GLIF approach is the possibility of adding didactics and supplemental material to the model. This feature is not present in Asbru 6.5b. Although it is not likely the absence of this information in the representation will change the actual guideline-model, it might very well influence the compliance of physicians to the guideline. Studies have shown that compliance increases when the rationale underlying the recommendations is known to the physician [22]. To prevent overloading users with information, the 'supplemental material' in GLIF can consist of references to either locally stored information or locations on the World Wide Web.

- Representation of sequences of decisions and actions.

PROforma and Asbru both use a hierarchical structure, where an ordering can be imposed on either the plans in the 'plan-task', or the sub-plans of the plan-body. This characteristic is notably different in GLIF, which uses pointers to create a directed graph. Although both techniques could lead to a correct representation of a guideline, the GLIF pointer structure does not allow the reuse of step(s) in other

guideline representations. In Asbru a plan-library is created, which allows a single plan to be part of different guidelines. Because of the ever increasing amount of guidelines, reuse of plans could lead to major reduction in modelling effort..

- Representation of eligibility criteria and criteria for making transitions between events.

In GLIF this aspect is very limited. Only a logical statement exist that can be evaluated as true or false. Based on those results a transition occurs either towards the destination attribute or the otherwise attribute. PROforma and Asbru have more extensive possibilities. The main difference between the latter two are PROforma's trigger condition which provides an opportunity to ignore all constraints, and Asbru's suspend and reactivate conditions.

- Guideline modelling process.

Both GLIF and PROforma divide the modelling process in two phases. First an intermediate model is created, which defines the raw structure of the guideline model in a graphical way. In the second phase, the intermediate model is extended with more detailed information, which results in the final representation. In PROforma there are tools to guide the designer in both phases, the graphical editor and the templates. In the Asbru modelling process there is no intermediate model, nor a tool to guide the designer. Whether or not the lack of such an intermediate model makes the representation exercise more difficult has to be seen. The end result of the modelling processes is rather difficult to read, probably even more so for physicians. A graphic intermediate model might help domain-experts determine the correctness of the representation.

PROforma is a formal language in a procedural sense, allowing direct execution (a very positive aspect) , yet it has a limited expressiveness. The GLIF version has several basic components that are only represented as strings or free text limiting computer interpretation. In General the Asbru language has the most extensive expressiveness, it is very declarative, and it is a very structured language. During the guideline modelling process PROforma offers significantly more support, as Asbru provides none.

3 Asbru in detail

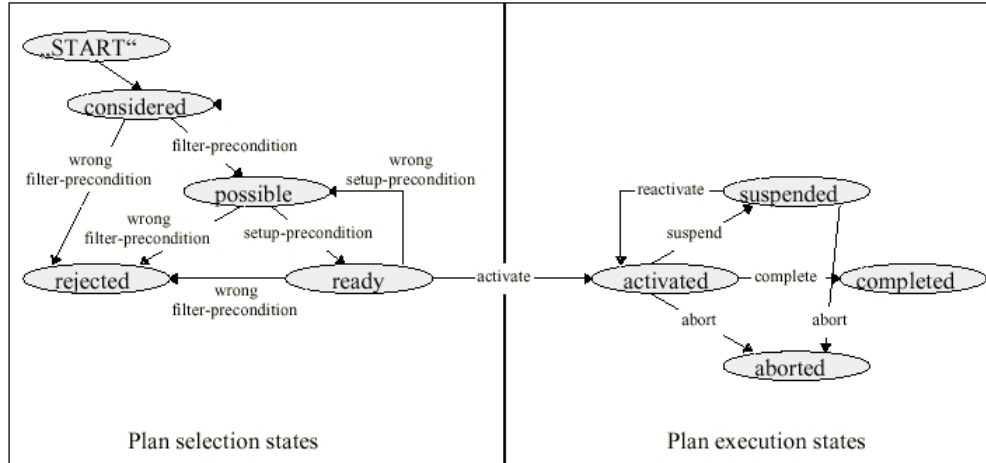
In paragraph 2.4 the background and goal of the Asbru language have been described, and the elementary concepts of the language were introduced. This chapter will therefore provide more detailed information on the structure of the language, plan-subplan relations, how the model is interpreted during execution time, and, in separate paragraphs, the concepts and their BNF notation.

In Asbru guidelines are modelled as a set of hierarchical skeletal plans.

Skeletal plans are “*plan schemata at various levels of detail, that capture the essence of the procedure, but leave room for execution-time flexibility in the achievement of particular goals*” [23]. The (medical-) knowledge of the guideline is then represented as a library of these skeletal plans. During runtime this means that single components of a such a represented guideline can be replaced based on the knowledge roles intention and condition: alternative plans with similar intentions can be recognised. The conditions of an alternative plan then determine whether or not it can be activated.

As mentioned the Asbru guideline representation is basically a plan library, and these plans have a certain hierarchical structure. It is this hierarchical structure that implies parent-child relationships between a plan and it's sub-plans. Plan states in Asbru are presented in figure 7. During execution a plan progresses through a number of these predefined plan-states.

Figure 7 Sequence of Asbru plan states



The golden rule in the parent-child relationship is that a child's plan-state can never overtake it's parent-plan's plan state. When a plan is executed, it is decomposed into each sub-plan (children), until plans are reached that can not be decomposed, called 'primitive plans'. These primitive plans usually perform interactions with users or external devices, such as electronic patient records.

All plans in Asbru (child or parent or both), are identified by a unique name (compulsory), and can be composed of a set of arguments, a time annotation, plan-body, conditions, intentions, effects and preferences, all optionally.

In the following paragraphs all of the concepts/components a plan can contain will be examined more closely. An example of the use of certain concepts and components is

presented earlier in figure 6, page 12. Since these concepts and components are defined in BNF, this notation is explained below in Figure 8.

Notation	Meaning	Notation	Meaning
::=	Is defined as	(expr)	Grouping expr in syntax
	Or (disjunction)	[expr]	Optional expr
Blank	And (conjunction)	[t1, t2]	Time/duration interval
< >	Non-terminal symbol	+	One or more
A...Z, a...b	Terminal symbol, <%string>	*	Zero or more

Figure 8 Backus Nauer Form

3.1 The Plan-body

Within the plan-body, the workflow of sub-plans or primitive plans is defined. The set of plans within the plan-body can be executed in *parallel*, *in sequence*, in *any order* or in a certain *frequency*. Besides this, an *if-then-else* statement can be included. There are three types of plans that can be distinguished: sequential, concurrent and cyclical plans. A plan body can contain only one type of plan.

A sequential plan

The specified set of plans are to be executed in sequence. For the sequential (parent-) plan to complete, all of its children have to be successfully completed themselves. In case of a do-some where not all subplans need to be completed successfully, the continuation condition specifies which child plans must be completed successfully for the plan to complete.

```
<sequential-plan> ::=
    (DO-ALL-SEQUENTIALLY (<assign-statement>+|
        <plan-activation>+
        <variable-plan-schema> |
        < variable-plan-schema-list>)) |
    (DO-SOME-SEQUENTIALLY (<assign-statement>+|
        <plan-activation>+
        <variable-plan-schema> |
        < variable-plan-schema-list>)) |
    [(CONTINUATION-CONDITION <plan-name>+)))]
```

In the BNF notation of a sequential plan, it's impossible to specify a continuation condition in case of a DO-ALL ordering. This is correct since a DO-ALL means all the plans must be completed to continue. It implicitly has a continuation condition that includes all plans.

A concurrent plan

Concurrent plans are executed in parallel. The BNF notation is similar to the BNF of sequential-plan, except 'SEQUENTIALLY' is replaced with 'TOGETHER'. Again the DO-ALL implies completion of all sub-plans before continuation, so a continuation-condition is unnecessary.

A cyclical plan

This includes one plan that can be repeated, and is defined by the operator EVERY. Temporal information that can be specified includes the starting time, the time-interval over which the plan is repeated, and the start time, end time and duration of a particular plan instance in a cycle. There are three ways the repetitions can be stopped: 1: by specifying an ending time point. 2: by means of the 'until condition', and 3: by specifying how many attempts are allowed using the 'times-attempts' argument.

```
<cyclical-plan> ::=
  (EVERY
    (START <start-time>) <plan-name>
    [(ARG (<plan-argument-value><arg-unit>)+)]
    [TIME-BASE (<time-range> <set-of-cyclical-time-points>|
      <variable-cyclical-time-annotations>)]
    [(COMPLETE <set-of-cyclical-complete-conditions>)]
    [(TIMES-ATTEMPTS <max-number-of-attempts>)]
    [(RETRY-DELAY (<min-delay>, <max-delay>))]
  END-EVERY)
```

An example that is often used to demonstrate the notation of a cyclical plan is: *"Administer 5 units of insulin every morning between 8AM and 11AM, starting with the first morning following the initiation of the plan"*. This would be written as: *(EVERY (START (FIRST (MIDNIGHT) after (ACTIVATED *self*)) (TIME-BASE [[8 HOURS, 8 HOURS], [11 HOURS, 11 HOURS], [_,_], MIDNIGHTS] administer-insulin 5) END-EVERY*.

Of course a number of abstractions is missing in the example above, like the reference point 'MIDNIGHTS'. The appropriate place to specify these abstractions is not clear.

3.2 Time annotation

In the medical domain time aspects are of crucial importance throughout almost the entire care process, and clinical data would be far less valuable without time-stamps. Unfortunately it's not always possible to plan or predict when something will happen, or when it will finish. This is called temporal uncertainty. As mentioned in the previous chapter, the Asbru language allows the representation of uncertainty in starting time, ending time and duration. It also supports the use of multiple timelines, by use of different reference points and different time units. The time annotation is written as:

```
<time-annotation> ::=
  ([Earliest Starting Shift, Latest Starting Shift],
  [Earliest Finishing Shift, Latest Finishing Shift],
  [Minimal Duration, Maximal Duration],
  REFERENCE)
```

Not only the time-shifts allow for the representation of uncertainty, also the reference point can have uncertainty by defining an uncertainty region. It is possible that one or more components of a time-annotation are unknown or undefined. If this is the case, the component in question is represented by an underscore '_'. Only the

REFERENCE is not allowed to be unknown. The example of a cyclical plan on the previous page contains an example of a time-annotation.

3.3 Condition

Conditions control the sequence of all the proposed actions described in the guideline. They 'govern when it's appropriate to perform certain actions' [10]. In the Asbru language they enable transitions from one plan state to another (see figure 7). Only when a complete condition becomes true, the plan is finished successfully. The seven conditions have already been introduced in paragraph 2.4, but will be covered more detailed in this section. The continuation condition is a special kind of condition, as it's appearance depends on the ordering of the plan body. Therefore the continuation condition has been covered in paragraph 3.1 instead of here.

Filter preconditions:

These are responsible for the transition of a plan from the plan state 'considered' to 'possible'. The filter condition can not be achieved through interventions. This means the filter condition has to be checked only once: if it is not applicable, it never will be. When the evaluation result is false, the plan is rejected. Rejected is a final state, the plan can not be reactivated.

Filter preconditions usually are constants (like gender), but can occasionally be variables that will be checked only once.

Setup preconditions

These are responsible for the transition of a plan from the plan state 'possible' to 'ready'. Only when a plan is possible, the setup condition checks whether or not it can and should be applied to a certain patient. Setup conditions may be achieved, and a failure will therefore not result in a rejection, but the plan remains in the 'possible' state, waiting for the setup condition to succeed, for example through the intervention of another plan..

Activate conditions

These are responsible for the transition of a plan from the plan state 'ready' to 'activated'.

The activate condition determines if a plan is activated automatically or manually after reaching the ready state.

Suspend conditions

These are responsible for the transition of a plan from the plan state 'activated' to 'suspended'. It creates the possibility to temporarily stop a plan, to be resumed at another time. The plan is interrupted, but not terminated.

Reactivate conditions

These are responsible for the transition of a plan from the plan state 'suspended' to 'activated'

Abort conditions

These are responsible for the transition of a plan from the plan state 'activated' to 'aborted'. When the abort condition is met, the corresponding plan will terminate unsuccessfully. A plan can be aborted not only when its state is active, but also while its state is suspended.

Complete conditions

These are responsible for the transition of a plan from the plan state 'activated' to 'completed'. The plan in question is successfully terminated.

3.4 Intention

The set of intentions models the aims of the plan. They are temporal patterns of actions or states that should be maintained, achieved or avoided, the three 'intention-verbs'. Some of these should hold during the plan (INTERMEDIATE), whereas others should be true after execution of the plan (OVERALL). Altogether this means four different types of intentions exist, to support tasks as critiquing or plan modification.

<set-of-intentions> ::=

```

    [(INTENTION: INTERMEDIATE-STATE <intention>+)]
    [(INTENTION: INTERMEDIATE-ACTION <intention>+)]
    [(INTENTION: OVERALL-STATE <intention>+)]
    [(INTENTION: OVERALL-ACTION <intention>+)]

```

<intention> ::=

```

    (<intention-verb> <temporal pattern> [<importance measure>])

```

The importance measure is defined as <%FLOAT>, and ranges from 0 to 1.. It describes how strongly the intention is required.

Since all four categories of intentions can be maintained, achieved or avoided, 12 different intention forms are possible (see paragraph 2.4)

<temporal-pattern> ::=

```

    <pattern> |
    (PLAN-STATE <plan-pointer> <plan-state-constraint> <time-range>)

```

<pattern> ::=

```

    <parameter-proposition> | <constraints> <pattern>+

```

<parameter proposition> ::=

```

    (<parameter> <value-description> <context> <time-annotation>)

```

As can readily be seen, an intention can be based on a certain parameter. This parameter can be a variable or an abstraction-function. The abstraction function plays an important role in Asbru. All abstractions are globally accessible, and can thus be re-used by different plans. Moreover it allows for qualitative parameters values. The parameter is related to a valid time annotation in a certain context. Context can be a

certain age-interval, or a certain (co-)disease, which influences the parameter. An abstraction can be context depending and changing, while the guideline needs not to be changed. As an example the following intention statement is modelled: *'the goal of the embedding plan is to keep the weight-gain of a pregnant woman in the ranges slightly-low, normal or high (abstractions), in the context of therapy for GDM type II during the time-interval, starting 24 weeks after conception, and ending at the time point of delivery'*.

(INTENTION: INTERMEDIATE STATE

(MAINTAIN STATE (mother-body-weight-gain)

(OR (SLIGHTLY-LOW NORMAL HIGH) GDM-TYPE-II

[[24 WEEKS, 24 WEEKS], [DELIVERY, DELIVERY], [_,_], CONCEPTION]]))

3.5 Effects

Effects can be used to describe the functional relationship between the plan arguments and measurable parameters, or to describe the overall effect of a plan on parameters. An example of a functional relationship could be the dosage of oral lipid-reducing substances like Atorvastatin, inversely related to the cholesterol level. The overall effect would be the decrease of cholesterol.

<set-of-effects> ::=

[(ARG-DEPENDENCY (<argument> <context> <parameter>
<relationship-function> <time-annotation> <likelihood>)+)]

[(PLAN-EFFECTS (<context> <parameter> <direction-of-change>
<time-annotation> <likelihood>)+)]

In a different context, a plan may have a different effect. All effects can have a likelihood, and a likelihood of 1 means it will certainly happen. Direction of change is defined as '`<symbol>|INC|DEC|NORMAL`', and the relationship-function is '`<symbol>|POSITIVE-MON|NEGATIVE-MON`'.

3.6 Preferences

This is used to express a preferred kind of behaviour of the plan. It is a constraint in the selection of a plan to achieve a given goal. As described in paragraph 2.4, strategy, utility, look-ahead-flag, select-method, resources are distinguished preferences.

When we take a closer look at resource-constraints, we can distinguish between PROHIBITED, RECOMMENDED, DISCOURAGED, OBLIGATORY resources, and the RESPONSIBLE-ACTOR. All the resource-constraints allow the specification of a time-annotation, specifying when a resource must be available. The responsible actor could be the DOCTOR, the NURSE or the PATIENT.

An example of a preference constraint is the modelling of a plan that *'should be applied with minimal treatment costs, and it prohibits the administration of penicillin'*.

(PREFERENCES

(UTILITY MINIMAL COSTS)

(RESOURCES PROHIBITED (PENICILLIN))

Preferences can play a role in the selections of appropriate plans, for example to make a choice when multiple plans have the same desired intention, or when a certain obligatory resource is not available at the hospital in which case the plan cannot be selected and an alternative must be found.

Although one could always explain more extensively the possibilities of Asbru, the level of detail presented in this chapter should suffice for reading and understanding the next chapters. When more details on the Asbru 6.5b syntax are desired, these can be found at <http://www.roomans.net/asbru/>

4 Guideline formalisation

To investigate whether the Asbru language is useable for representing medical guidelines, a medical guideline of the American Association of Pediatrics on hyperbilirubinemia [25] has been modelled using the Asbru language. This guideline was the main guideline of the Sisyphus project we planned to participate in. More information on this project will be presented in section 4.1.2. By modelling the guideline information was obtained on the guideline aspects that can be modelled in Asbru theoretically, and on the modelling effort required to model these aspects in practice. Prior to the actual modelling of the guideline, more knowledge on the guidelines subject of hyperbilirubinemia was needed. Both knowledge acquisition and modelling are considered prime activities, a division that is continued within the 'Materials and Methods' and 'Results' sections. The results of both activities are divided into *products* and *lessons learned*. After the results are presented they will be discussed in section 4.3, followed by the conclusion in relation to the posed research question in section 4.4.

4.1 Materials and methods

Each of the following two subsections, on knowledge acquisition and the modelling process, starts with a description of the materials used, followed by a description of the methods.

4.1.1 Knowledge Acquisition

One of the ways to increase one's knowledge on a subject is to perform a literature study. After this literature study was performed [26-29] remaining questions on the guideline terminology were answered by the domain expert.

Within the Sisyphus project [24] a number of test-cases for the guideline were offered, meant to help during the development and evaluation of the application. In collaboration with another project the planned application was a critiquing system. The sequential actions proposed by the domain-expert, based upon the cases, could function as the input data for the critiquing system. Since literature had already provided us with the theoretical knowledge on the subject, we decided to also use the cases to increase our understanding of the way things would work in reality. Furthermore, these answers could provide us with information on the intentions of actions and decisions, information not completely present in the guideline.

Knowledge Acquisition Material

Altogether 13 cases were provided by the Sisyphus project of which we used the first 8, keeping the other five in reserve. The following case serves as an example of the Sisyphus cases.

Case 1:

"Admission of a term icteric newborn. Gestational age is 40+2 weeks. Spontaneous vaginal delivery. On day 2 after birth laboratory data show a

significantly elevated total bilirubin of 17.8 mg/dl. Blood group of the mother: O, Rh positive, antibodies negative. Blood group of the child: A, Rh positive, negative direct Coombs test."

In a case such as this a number of evaluating questions have already been answered, and certain actions have already been performed. For example the blood types of both mother and child have been measured, and the total bilirubin level has been measured on day two. Since we wanted to be able to critique the entire diagnostic and evaluation phase, we decided to limit the initial information presented to the domain-expert.

*Admission of a term icteric newborn, . Gestational age is 40+2 weeks.
Spontaneous vaginal delivery, day two: 17.8 mg/dl TSB.*

All the other information was presented when the domain expert performed the specific action, e.g. when the domain expert wanted to perform blood typing, the known data was presented.

Knowledge Acquisition Method

After the domain-expert had had enough time to become familiar with the guideline, a period of time was planned before the knowledge elicitation started. Knowledge elicitation was not started immediately, to create a more realistic situation in which the domain expert has not recently studied the guideline. A case study method was chosen for the knowledge elicitation, called 'Forward scenario simulation', which provides information on the followed procedures and the reasons behind them [31,32]. Since the interview concentrated on cases, the expert was less likely to digress, making the answers more coherent [30]. In knowledge eliciting interviews, it's sometimes difficult to know which parts of the dialogue are important, therefore all details of the interview must be kept. [30] It is for this reason that all the interviews were taped, and later written out in full. These transcripts were used to create a schema of each case, identifying the actions and decisions made by the domain-expert, combined with their corresponding intentions. This 'schema with intentions' was then presented to the domain-expert, to further refine the results. The interview was conducted in two sessions of approximately one and a half hours each, the second session took place one week after the first.

4.1.2 Modelling process

After the knowledge acquisition was finished the actual modelling process started, which should yield the most important results for the answering of the research question. The most recent version of Asbru was version 6.5b, of which a document containing the syntax in BNF was available as mentioned in chapter 3. Both the guideline material and the modelling method are presented, though in separate sections.

Modelling process material

The clinical guideline chosen as the subject of the modelling exercise is the guideline for the 'Management of Hyperbilirubinemia in the Healthy Term Newborn' from the American Academy of Pediatrics (AAP) [Appendix B]. This was the central guideline of the Sisyphus Project IV [24], which we planned to participate in. The Sisyphus project IV was meant to be an international collaboration initiative, aiming to produce a quantitative and qualitative assessment of knowledge acquisition and knowledge engineering methods. To this end a system had to be developed that supported medical professionals that were not necessarily specialist in neonatology to manage causes of jaundice as specified in the inclusion criteria of the hyperbilirubinemia guideline. The mentioned guideline would function as a starting point and a compulsory source. Unfortunately, for reasons unknown to us, the Sisyphus project never left the initial phase.

The domain of this guideline is neonatal medicine, and although this is not explicitly stated in the guideline itself, it is meant for use by the physician. The guideline is divided into an evaluation part (largely diagnostic), a treatment part and an appendix. When compared to the five guideline types, identified by the Institute of Medicine (paragraph 2.1), the guideline would be considered a combination of type 2,4 and 5:

- *Diagnosis and prediagnosis management of patients* (the evaluation part).
- *Appropriate use of specific technologies and tests as part of clinical care* (the appendix).
- *Guidelines for care of clinical conditions* (the treatment section).

Using an alternative characterisation based on the format in which the guidelines knowledge is presented, the hyperbilirubinemia guideline would be considered a combination of:

- *Narrative text* (the main part of the guideline).
- *List* (guideline table one and table three).
- *Table* (guideline table two).
- *Flowchart* (the guideline algorithm).

Including the introduction and references the size of the guideline is fourteen single sided A4 pages. One of these fourteen pages contains the evaluation information, and the treatment segment takes a bit less than two pages. The Appendix covers one and a half page, and five pages contain the tables and the algorithm. This means the actual guideline information is presented on less than ten pages.

After the first steps in the modelling process where taken, it became apparent that the hyperbilirubinemia guideline contained little information on the time-annotation aspect. A number of guidelines were investigated to find temporal information as practice material, because the time-annotations were considered an important aspect of Asbru's possibilities and therefore an important practice subject. Phrases containing temporal information from the Academic Medical centre's guideline for post-operative treatment of patients with cardiac diseases were used to obtain experience with the modelling of temporal information since the kind of temporal information they contained was found quite commonly throughout the investigated guidelines.

Modelling process method

Once the knowledge acquisition was completed, the modelling process itself was started. Possible separate plans were identified at a high level, such as a diagnostic plan and a treatment plan. For every plan or sub-plan, plausible sub-plans were identified until no further decompositions could be made, a top down approach. After the separate plans were identified, the Asbru components were modelled subsequently in all plans, starting with the high-level plans and continuing with a top down approach towards the leaf-plans.

The Plan body was the first plan-component to be modelled, defining the workflow of the plans. This ordering of the plans sometimes imposed a possible continuation condition (e.g. DO-SOME), which was modelled together with the complete conditions, after which the other conditions such as filter and setup were modelled. The time-annotations, effects, preferences and intentions were subsequently modelled in all plans, marking an important phase in the modelling process, in which all components were present.

Because of the unfamiliarity with Asbru, a direct notation of the guideline in Asbru syntax (as in Chapter 2.4, figure 6) decreased the readability. A decreased readability led to a loss of overview, especially as more components were added and the model size increased. Every time a new component was added to the model an iterative process started, during which the understanding of that component was increased. Sometimes the model changed because of this better understanding, sometimes because it became apparent new plans were needed, and sometimes existing plans needed to be split or even removed. As the model increased in size, the overview had to be increased. In order to do this an intermediate notation was created which enhanced readability, and thereby supported overview.

4.2 Results

First the resulting products of the knowledge acquisition process and the lessons learned from this process will be presented, after which the same will be done for the modelling process in 4.2.2.

4.2.1 Knowledge Acquisition results

Knowledge acquisition generally helps the designer of the guideline model to increase his or her understanding of the subject involved. It is near impossible to present this increased understanding on paper, but the more tangible results are shown in this paragraph. As mentioned 'products' and 'lessons learned' are covered in separate sections.

Products

Based on the transcripts of the case-based interviews with the domain expert, the actions and decisions made by this expert were converted into schematic figures. Figure 9 is an example of such a schematic figure. Some of the domain-experts

actions where meant to obtain certain data. Whenever this data was present in the case the correct information was presented. Some data however was not available. In these circumstances possible answers where considered. Questions where posed on the informative value of these different answers, though the case was continued by assuming the value of the lacking data as 'normal'.

Only explicit decisions were modeled as decisions, the decision to determine the ratio of direct / indirect bilirubin was modeled as the action of determining this ratio, the arrow indicating the implicit decision.

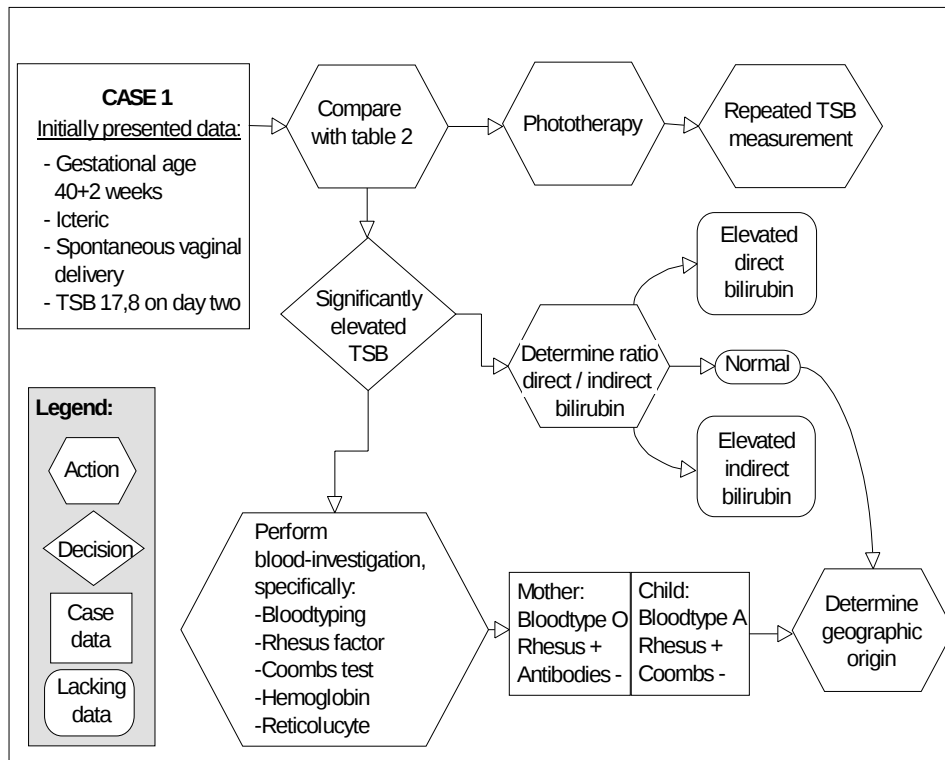


Figure 9 Actions and decisions of domain expert based on case 1.

After the actions where identified, a check was made whether or not their corresponding intentions were mentioned in the interview. The transcripts, the schematic figures and the list of actions with their corresponding intentions were presented to the domain expert for verification. The list of combined actions encountered in the cases, together with their corresponding intentions (figure 10) does not contain all actions present in the original guideline, and some actions made by the domain expert were not present in the guideline. The list was however very relevant to the testing of the critiquing system that would use the model of the guideline and the obtained case data as input. The availability of the intentions, corresponding with the domain experts actions would allow the system to critique based upon these intentions. When a physician would perform an action, different from the action described in the guideline, the system could check for matching intentions. Of course for this to work, the plans of the guideline model would need to have their corresponding intentions available as well. Thought was the list of combined actions encountered in the cases, together with their corresponding intentions could be used to compensate for the absence of information on intentions in the guideline. This proved not so easy, as the guideline model often needed a higher level of intentions. Low level intentions were often related to low level, or

primitive, plans. These primitive plans, such as ‘ask’ or ‘obtain’ can only be connected to an intention when they are turned into a plan, containing the primitive plan in the plan body, creating an artificial parent plan for each primitive plan. This is a very discursive approach, which is why typically a higher level intention in the primitives original parent plan was needed.

Action	Intention
Total Serum Bilirubin measurement.	1. Determine extent of hyperbilirubinemia. 2. Determine (preliminary) therapy.
Repeated TSB measurement.	1. Determine effect treatment. 2. Determine presence hyperbilirubinemia. 3. Determine therapy.
Ratio direct / indirect bilirubin.	Determine presumable cause hyperbilirubinemia: Indirect elevated: hemolytic possibility. Direct elevated: pathologic possibility.
Blood typing mother and child.	Investigation of blood type antagonism as cause hyperbilirubinemia.
Determination rhesus factor mother and child.	Investigation rhesus conflict as cause hyperbilirubinemia.
Measure amount reticulocytes.	Investigate possibility hemolytic disease.
Determine enzyme deficiencies.	Investigate possibility error of metabolism.
Direct Coombs test.	Determination of probability blood type antagonism or rhesus conflict.
Hemoglobin test.	1. Determine presence and seriousness hemolysis. 2. Determine necessity of exchange transfusion.
Haptoglobin test.	Investigate possibility hemolytic disease.
LDH test.	Investigate possibility hemolytic disease.
Determine shape erythrocytes	Investigate possibility hemolytic disease.
Geographic origin child and parents.	Investigate probability G6PD deficiency.
Anamnesis question cesarean section	Possibility of swallowed blood, causing jaundice.
Anamnesis question breastfeeding	Possible cause of the jaundice
Compare TSB-level with table 2 from guideline.	Determine appropriate therapy.
Prescription of photo-therapy.	1. Lower TSB value 2. Avoid detrimental effects high TSB.
Observation.	Anticipate increasing hyperbilirubinemia or symptoms.

Figure 10 Identified actions and their corresponding intentions.

Lessons learned

As expected the literature study proved to be a very useful activity. It provided the necessary information to the basic understanding of the guideline. This basic understanding proved necessary during the interview with the domain expert based on the cases. Whereas the cases helped to structure the interview, the understanding of the domain helped in understanding the answers and asking the right kind of questions. The interview based on the cases further increased domain knowledge, and provided the data that was required as input for the critiquing system. The utility of the cases as a tool for the acquisition of intentions for the guideline model is

questionable. When the cases, and more specifically the domain-experts solutions to these cases cover (almost) the entire range of actions present in the guideline the results would be very useful, though the time expenditure would remain rather large. The intentions required by the guideline model that were not obtained during the case-interviews could probably be acquired by asking questions to the domain expert based on the plans in the guideline model and in very close relation to the original guideline. Instead of using cases it might be more useful to acquire all necessary intentions this way. Another negative aspect of the case-interview is it proved very hard to separate the knowledge obtained in these interviews (not to be modelled) from the knowledge obtained and present in the guideline (to be modelled). The recommendable ordering for knowledge acquisition for the modelling of guideline based on this experience would be:

- Literature study
 - Posing remaining questions on the guideline material to domain-expert
- Followed by an iterative process consisting of:
- The modelling exercise itself
 - Interview with the domain-expert to obtain knowledge on specific gaps (such as intentions) or necessary data (for example based on cases).

4.2.2 Modelling process results

An important result of this study is the actual model of the hyperbilirubinemia guideline, which can be found in the appendix [B]. This model is based on the Asbru 7.2 syntax. As mentioned in the earlier chapters, when this study started the latest Asbru version was 6.5b. This is why after the knowledge acquisition exercise, at first a guideline model was created based on the 6.5b syntax. The acquired experience resulted in a number of recommendations, part of which were presented to the Asbru team in Vienna. A number of changes were made to the Asbru syntax, which finally resulted in Asbru version 7.2. The initial 6.5b model was then redesigned according to this 7.2 version. The comparison of the 6.5b and 7.2 model provided information on the impact of these changes. The resulting products of the modelling exercise are besides the guideline model itself the template that is used. The lessons learned section contains a list of recommendations on Asbru, based on the modelling experience.

Products

The lack of a good intermediate model was definitively felt. A graphic intermediate model might help domain-experts determine the correctness of the representation and ease the modelling process itself, especially since the Asbru model of a guideline has a tendency to become rather large. The creation of a tool to create such an intermediate model is therefore strongly recommended. To ease the modelling process in spite of the absence of an intermediate model, a template structure was used to store the available guideline information in (figure 11).

PLAN number	Unique name	ARGUMENTS:	
PREFERENCES			
INTENTIONS			
CONDITIONS	Type	details	
EFFECTS			
PLAN-BODY	DO..... Sub-plan (Continuation specification)		
RETURN VARIABLES			
COMMENTS			

Figure 11 Template-like structure, able to contain a single plan.

Within this intermediate model a plan is presented as a unit. This ‘plan-unit’ has a unique name (as it has in Asbru) but also a hierarchical number to indicate parent-child relationships.

The template proved a great initial way to learn working with the modelling language without losing track in the model itself. However as the number of templates (plans) increased overview diminished again. A filled template can be seen as a kind of semi-Asbru, since the information in the template is according to the Asbru syntax. This makes it easier to make the transition of the model to real Asbru, whether LISP-like (version 6.5b) or XML (version 7.2).

Lessons learned

Most of the lessons learned resulted in recommendations that were made in regard to Asbru 6.5b. These recommendations are some of the key contributions of this thesis, and consist of adaptations, extensions, or choices to be made. Besides more general recommendations, of which quite a few concern the use and behaviour of variables, others are of importance to the kind of knowledge that can be modelled in Asbru, the reduction of the model size or the reduction of the modelling effort.

Data-structures

Local variables or global variables.

A choice had to be made whether to make use of local variables which are accessible only within a single plan and whose results are transported to other plans when necessary, or to use global variables that are accessible by all plans. This choice needed to be made since both options were allowed by the syntax. The easiest way from a modelling point of view is the use of global variables, there are however a few reasons in favour of the local variant. One of the main benefits is the effect that plans can be modelled independently and therefore be more reusable than with global variables which must be declared somewhere in the plan-library. Another benefit is the increased knowledge on the behaviour of variables, and where they can be changed. Drawback of the use of local variables besides the increased modelling effort is the appearance of plans needed to assign certain values to variables, instead of appearance based on guideline information.

It is recommended to at least create the possibility for local variables, preferably done in combination with the possibility for global variables. In the 6.5b model only local variables are used, primarily because of the increased reusability of the plans. The drawback of the creation of extra plans can be avoided by increasing the possibilities of the If-Then-Else statement in the plan body, which will be explained in the section on ‘Representation of sequences’.

Arrays

A way needs to be found to deal with table structures in guidelines. Even the table in the hyperbilirubinemia guideline which consists of only two dimensions (Age and TSB level) increased the model size significantly, as for example filter conditions relating to the table took forms like: 'IF ((Age>24h AND Age=<48) AND TSB>12) OR ((Age>48h AND Age=<72) AND TSB>15) OR ((Age>72h) AND TSB>17) THEN "consider phototherapy", ELSE IF (et cetera for the other treatments). Of course every field of this two-dimensional table could be put in an abstraction to limit the increase of model size, though it is debatable whether it is a good idea to hide all this medical information from the model itself. Besides this, imagine what happens when arrays with more than two dimensions need modelling. This would lead to an immense increase in either the size of the model, or the amount of the needed abstractions.

Type declaration of variables.

A type declaration is needed in Asbru, and is not yet available. Besides deciding on where global or local variables should be declared and in what syntax this should be done, what types can be assigned to a variable is very important as well, and it is recommended to allow the possibility of sets. This would create the possibility to for example declare a type called 'Bloodtype'. 'Bloodtype' is actually a set of possible blood types, and thus consists of 'A', 'B', 'AB', 'O' and maybe 'unknown' or '?'. No other values can now be assigned to any variable of type 'Bloodtype', thus avoiding the input of impossible data.

By limiting the allowed number of values that can be assigned to a variable to for example 'true' and 'false', the variable could be used in a filter-condition stating 'if <this variable> = 'true' then <something>'. Without the reduction of answers to 'true' and 'false', a physician might assign the value 'yes' to the variable. The filter condition would not trigger the <something> event. In this case the typing of variables avoids the necessity of modelling filter conditions like "if <this variable> = 'true' or 'yes' or 'positive' or 'correct'", which could never be completely exhaustive.

Initial values of variables and parameters.

Besides the type declaration necessity of the previous recommendation, a decision has to be made on initial values of variables and parameters. Either a standard value such as 'undefined' is chosen, or a way to assign initial values has to be made.

The first solution might be preferable for global variables and parameters, since these are declared in the plan-library and can be reused. A standard initial value would make it easier on the modeller who would like to use already existing variables or parameters. 'Undefined' would be a good default value, as it provides clarity on the difference between a test or plan that hasn't been done (undefined) and a test or plan that has been done but didn't yield result (for example 'unknown'). For local variables one could adopt the same standard initial value, or enable the modeller to assign a different initial value, for example near the declaration of such a local variable.

Assign return value to a variable.

Sometimes an invoked plan resulted in a return value. In most of these cases a primitive 'ask' plan requires information, either from user input or from a database. It is not possible to identify the variable this return-value is stored in. This is a small but nevertheless important extension that is needed.

Abstractions

It is not possible to declare an abstraction in Asbru 6.5b, though they are used for example in the reference of temporal annotations. This makes it likely that the possibility of declaring abstractions in Asbru was already planned for future development, and this is applauded. Besides the need for abstractions from a modelling point of view, abstractions ease the modelling process significantly once they are defined, and increase readability of the model. Although abstractions are very useful, there are a few drawbacks. When a plan is reused in which an abstraction is used, this abstraction should be reused as well. This means a modeller needs to know which abstractions are already available within the plan-library, and how they are defined. The more aspects of a plan have to be reused, the smaller the chance that reuse is possible, especially when for example filter conditions are included. Another reason abstractions might not be reusable very often is the fact a lot of abstractions are defined by a guideline though there's no consensus on the definition. Take for example the hyperbilirubinemia guideline which contains TSB values that indicate when a certain treatment is appropriate. Some abstractions are based on these values but the values are rather arguable, making it unlikely the abstraction can be reused in another guideline model, thus making it difficult to reuse the plans in which these abstractions are used.

Functions

Sometimes mathematical operations are needed, for example to determine the increase rate of certain parameters over time. This is a very important aspect that remains unclear. Since the data itself can be given a time-stamp, only the mathematical operation is lacking.

Intention

The hyperbilirubinemia guideline very rarely stated the intentions behind recommended actions and therapy decisions, and in those cases where an intention could be identified it often was stated implicitly. In spite of the absence of most information on intentions, one should not conclude from this a guideline modelling approach might not need the possibility to model intentions. Especially when an electronic guideline representation must function within a critiquing system or decision support system in an intelligent manner, intentions could provide essential additive information. A decision support system could for example provide alternative actions based on the intention that has to be fulfilled, and whenever a physician chooses an action that differs from the recommended guideline action a critiquing system could compare the intention of both actions to make it's critiquing more refined. This is the reason intentions can be modelled in Asbru, and this is why an effort has been made to identify the intentions of all plans within the model regardless of lacking guideline information. This effort lead to the problem of achieving an intention. This is an important aspect, since it creates the possibility to check for the verification of an intention (for example an intention is to know whether or not a pathologic reason exists. Once all necessary tests are done, an answer is found (either yes or no) and the plans intention is fulfilled.

To allow verification of an intention, a variable had to be created, and based on the information resulting from the plan, a value had to be assigned to this variable. Sometimes it didn't matter what this value was just as long as a value was assigned to the variable (var = NOT 'undefined', or var = known) , in other cases a specific result was required to fulfil the intention. In the intermediate model a solution was proposed in which values are assigned to variables which are present in the intention (see the example below).

PLAN Anamnesis-Hemolytic-DiseaseIntention:

Achieve Overall State:

Possibility-of-Hemolytic-disease = known ANDPossibility-of-inherited-disease = knownPlan-Body:

DO-ALL-SEQUENTIALLY

PLAN anamesisquestions

Possibility-of-Hemolytic-disease = noPossibility-of-inherited-disease = no

PLAN evaluate-hemolytic-disease

PLAN evaluate-inherited-disease

PLAN anamesisquestions

DO-ALL-ANYORDER

Ask Family-history

Ask Ethnical-origin

Ask Geographic-origin

Ask timing-of-appearance-jaundice

PLAN evaluate-hemolytic-disease

Filter Condition

(Family-history = yes) or

(Ethnical-origin = yes) or

(Geographic-origin = yes) or

(timing-of-appearance-jaundice = yes)

Plan-BodyAsk Possibility-of-Hemolytic-disease**PLAN** evaluate-inherited-disease

Filter Condition

ethnical-origin = yes

Plan-BodyPossibility-of-inherited-disease = yes

The main drawback of this solution was the enormous increase of plans, since a large number of plans were necessary to evaluate results of other plans. This problem existed because an If-Then-Else statement was only allowed in the plan-body to activate plans, not to assign values to variables. This aspect will be covered more extensively in the paragraph concerned with 'Representation of Sequences'.

Temporal information

As mentioned in 4.1 a number of guidelines were screened for temporal information as the hyperbilirubinemia guideline unfortunately contained little temporal information. The little temporal information that was part of the guideline (e.g. Age child) needed modelling in an abstraction rather than modelling as a time annotation. The temporal information found in the Academic Medical centre's guideline for post-operative treatment of patients with cardiac disease was found quite common as it was present in most of the investigated guidelines. The phrases containing the temporal information and their corresponding Asbru syntax are presented here.

The statement *“Do not administer pump-blood more than 6 hours after surgery”*, is more generally present in guidelines as:

“Do not start (PLAN_A) more than (temporal duration) after (Plan_B)”.

This kind of temporal information can be modelled as follows: Introduce a parameter (e.g. Surgery) whose state is set as soon as Plan B (Plan Surgical-operation) is started, and let the executor of the guideline monitor this parameters state. In Asbru syntax this would be written as:

```
PLAN_A
( FILTER PRECONDITION
    (State (parameter) started * ( [_,6HOURS], [_,_] , [_,_]
        (ACTIVATED FIRST-INSTANCE PLAN_A) )
    )
)
```

Another kind of temporal information that was present in more than one guideline was a cyclical statement, in this case *“Administer nebulous Fluimucil/Ventolin every three hours”*. More generally this could be written as: *“Execute (PLAN_A) now, and every (temporal duration) thereafter”*. In Asbru syntax this could be written as:

```
(EVERY
(START *NOW*) PLAN_A
[(TIME-BASE ([_,_] , [_,_] , [_,6HOURS] [_,_] , 6HOURS) )]) ]
END-EVERY)
```

The time base can also be modelled as (TIME-BASE Every-6-Hours), where every-6-Hours is a variable-cyclical-time-annotation, which is a lot easier to read.

The most common temporal information was information on the frequency of drug intake. For Example *“administer 0.25 mg of Digoxine every 4 hours after Ouabaine has been administered until 1,5 mg has been administered”*. This is actually a combination of a filter precondition and the cyclical action of administering a certain drug. The Asbru code of this phrase is more or less a combination of the two examples above:

```
(FILTER PRECONDITION
    (PLAN-STATE FIRST-INSTANCE (administer (Ouabaine,_,_))
        COMPLETED *)
(EVERY
(START
    Administer (Digoxine, 0,25mg)
```

```

(TIME-BASE (Every-4-Hours))
(COMPLETE (UNTIL-CONDITION Digoxin-Level 1,5 * *) )
END EVERY)

```

Instead of 'Every-4-Hours' also the TIMES-ATTEMPTS=6 ($6 \times 0.25 = 1.5$) possibility would have limited the intake to 1,5 mg overall. It will be clear that these examples do not cover all of the temporal information that can be modelled by Asbru. The temporal possibilities of Asbru are quite extensive and could not all be investigated, since adequate study guideline material that covered all these possibilities proved hard to find. It seems that for the majority of guidelines Asbru's temporal possibilities are over-extensive, and this makes the modelling of temporal information less transparent because of the many different options. However, when certain temporal information is rare this does not imply the possibilities to model it don't have to exist. Instead of limiting Asbru's possibilities it might be a better option to aid in the modelling process of temporal information, such as providing examples of more frequently present temporal annotations. The lessons learned from this are the fact there is sometimes more than one way to model the same information, and Asbru's possibility for the modelling of temporal information is very expressive and extensive.

Representation of sequences

Another thing that was concluded in the comparison of paragraph 2.5 was the difference of the representation of sequences between especially Asbru and GLIF. The pointer structure GLIF uses eliminates almost all possibilities to reuse a 'step' of a guideline, but is very easy and intuitively to use. The Asbru approach of representing sequences offers possibilities to reuse plans. Besides this it didn't prove too hard to model sequences in Asbru, making this approach to the modelling of sequences preferable. Still a recommendation to extend Asbru on the aspect of modelling sequences is made.

The Asbru approach assumes it is known which plans have to complete, either in a DO-ALL in which all plans must complete, or a DO-SOME when the necessary plans are placed in the continuation condition. Sometimes in medical practice this kind of information is not available. Imagine there's a list of tests:

Plan List-of-tests

DO-??-??

Plan test-1 → Filter condition: Bloodtype=A

Plan test-2 → Filter condition: Patient-Age>65

Plan test-3 → Filter condition: Pain-on-chest=true

When a DO-ALL is used in the above example, a patient who's 50 years old, has blood type A and pain on his chest will cause the parent plan to fail, because plan test2 does not complete. When a DO-SOME is used, what kind of continuation condition would be appropriate? This can't be said, since depending on their filter conditions either one, or none, or all have to be completed. It is therefore the introduction of a new concept called TRY-ALL is recommended. The TRY-ALL would be completed when all plans where the filter condition is passed are completed, creating a kind of dynamic complete condition.

Besides this TRY-ALL aspect, another recommendation concerning the ordering of plans was made. In Asbru 6.5 it is not allowed to use more than one kind of plan type within a plan. In practice it became apparent that it is not unusual for a plans to need more than one kind of plan typing.

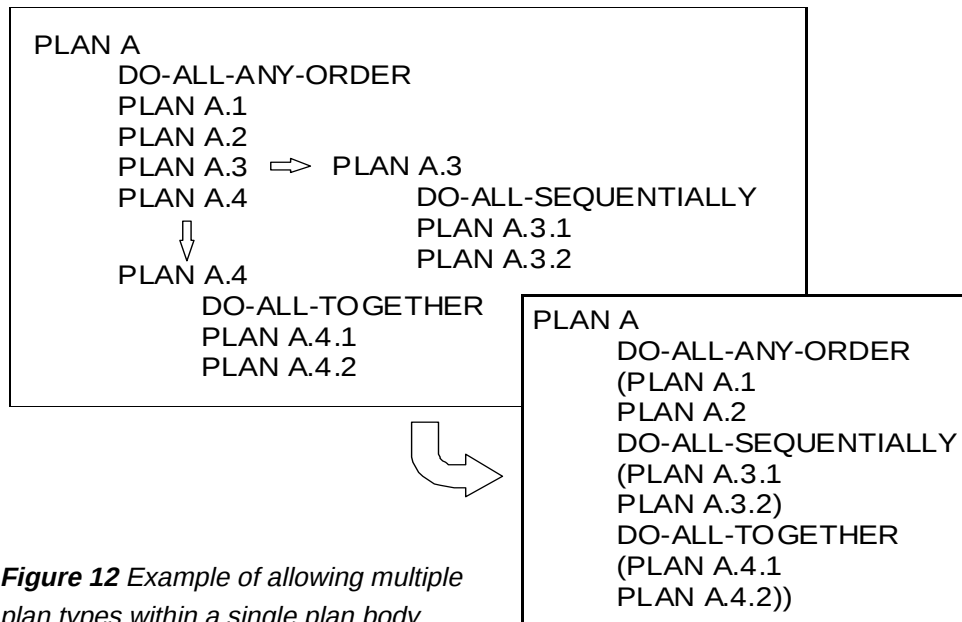


Figure 12 Example of allowing multiple plan types within a single plan body

As is illustrated in figure 13, one of the benefits of allowing multiple plan types within a single plan body is the reduction of plans. Imagine a plan that contains a number of sub plans, some of which should be executed sequentially or concurrently but it doesn't matter whether the sequence is started prior to the concurrent plans or not. In Asbru 6.5b this means a number of plans have to be created that lack medical information and have no real intention, they only exist for a modelling technical reason (PLAN A.3 and A.4 in figure 13). Besides the reduction of plans which increases the surveyability of the model, using multiple plan types allows the modelling of plan content based on the guideline information. This way the model can stay more true to the guideline it is based on. Drawback of this solution is an increased complexity of the remaining plans, which can decrease readability. Positive side-effect of changing the syntax this way is it would allow the use of multiple If-then-else statements in the plan-body.

The if-then-else statement is rather limited in asbru 6.5b. It can be used to activate a plan when a certain temporal pattern becomes true. It cannot be used to assign a value to a variable or parameters, activate a plan based on the values of certain variables or parameters nor can the else statement contain another if-then-else statement. It is recommended to expand the if-then-else statement with these possibilities. An example of the necessity of these expansions is plan 1.4.1.1 called 'Anamnesis abnormal signs' from the model available in appendix B. Here the if-then-else statement is used multiple times to evaluate values of variables or parameters to assign a value to another variable or parameter. In the Asbru 6.5b model it was necessary to create two extra sub-plans to evaluate the values of certain variables or parameters in their filter conditions, and these plans needed to be executed after the

other sub-plans were executed (thus in sequence). As these other sub-plans were of the type 'ANY-ORDER', first an extra plan had to be made since multiple plan types were not allowed. Obviously this way of modelling not only created extra plans that did not find their roots in the guideline, it also increased the modelling effort (see figure 13).

Without multiple plan types and without IF-THEN-ELSE expansion :	Allowing multiple plan types and expanded IF-THEN-ELSE statement:
PLAN A Intention: Known(X) Abort : X=true Plan-Body DO-ALL-SEQUENTIAL Plan A.1 Var X=false Plan Evaluate-A	PLAN A-new Intention: Known(X) Abort : X=true Plan-Body DO-ALL-SEQUENTIAL((DO-ALL-ANYORDER Ask 1 Ask 2 Ask 3) If Ask 1=yes OR Ask 2=yes OR Ask 3=yes Then X=true Else X=false)
Plan A.1 DO-ALL-ANYORDER Ask 1 Ask 2 Ask 3	
Plan Evaluate-A Filter: Ask 1=yes OR Ask 2=yes OR Ask 3=yes Plan-Body Var X=true	The existence of plans A.1 and Evaluate-A is not based on the guideline but on modeling technicalities. Plan A-new consists of all information belonging to that plan.

Figure 13 Example of Nesting Impact.

Property of the nesting of plans is the increased complexity of remaining plans, and a difference in the way the model works. In the original situation when all asks would have the value 'no' assigned to them, the plan evaluate-A would fail, causing plan A to fail. In the situation described by plan A-new, A-new would complete successfully with X=false. This is actually a better way, as the intention of plan A is Known(X). When all asks have the value 'no', this means X=false, which should mean the intention of plan A is reached and the plan should not fail. The increased complexity of the plans might reduce readability and overview (though limitation of the number of plans might increase both), and this increased complexity might cause a possible reduction of reuse for those plans. This is not necessarily a problem. In case of the if-then-else statements mostly 'evaluating' plans are eliminated. These plans, evaluating resulting values from the parent plan are very strongly connected to the parent plan, and would very likely be necessary every time the parent plan would be used. In such a case the condensation of the information in a single plan would actually ease the reuse of the parent plan. However a reuse problem might occur in the cases where multiple plan-types are used in a single plan. This should therefore only be used when all the plans involved are tightly related to each other and would

reasonably all be reused when the parent plan would be reused. If this is not be the case there is reason to model only one plan-type in a single plan body, like the way it is obligatory in Asbru 6.5b.

Didactics and supplemental material

Quite a lot of additional information from the hyperbilirubinemia guideline could not be modelled within the possibilities of Asbru 6.5b. Examples of such additional information are the prevalence of jaundice, information on the applicability of the guideline, information on phototherapy (e.g. commonly used lamps and intensities), and references to literature. These are very different kinds of information. Some of it is information that surrounds the actual guideline, like questions that are still being debated, other subjects are not used for recommendations since standardised methods for them are non-existing, though they still can be very informative like the additional phototherapy information. Maybe this kind of information can be made available in text, for example accessible by a button on a user interface. The references to literature or WWW-pages should however be placed near the appropriate recommendations. Though these kinds of information do not directly influence the care process, they are nevertheless important aspects of a guideline. They can increase the usability of guidelines for educational purposes [4] and might increase compliance to the guideline, as this information can be used to explain the rationale of the guideline [35,36]

Design Patterns

A design pattern can be seen as a standard model for standard modelling situations. This way the standard model only needs some minor adaptations depending on the specifics of the situation. One can identify two different kinds of design patterns:

1. Design patterns could be formulated for modelling situations present in many guidelines. In such a case they can be viewed as examples of preferred modelling strategies thereby assisting the modeller and help to increase the quality of the model itself. They will be referred to in the discussion as 'design patterns'. These design patterns would not have an impact on the modelling language itself as they aren't an extension to asbru-language (like for example the TRY-ALL suggestion), which makes their introduction a lot easier. The creation of this kind of design patterns will also initiate discussions on preferred modelling strategies in specific cases, creating a sort of jurisprudence which is not yet present.
2. Design patterns could also be formulated for the modelling of situations often present within one guideline. In this case they could be designed to be used as a kind of subroutine formulated by the guideline modeller, which will be referred to further on as 'subroutines'. In contradiction to the other kind of design patterns

this possibility will be a lot harder to introduce, since a new aspect has to be introduced into the modelling language itself. Their main benefit would be to ease the modelling process of guidelines in which certain modelling situations are repeated more often. Besides this they would limit the size of the resulting models.

The possibilities for the first kind of design patterns are rather extensive, as for all the situation occurring in guidelines a preferred modelling strategy can be created. Examples could be control structures or certain medical knowledge patterns like the temporal annotation for the prescription of a drug or the evaluation of an anamnesis. The use and creation of subroutines would greatly depend on the guideline in question.

Short summary of ‘Lessons learned’ from the modelling process section.
Recommendations
- Define the option to use both local variables as global variables, yet try to focus on local variables whenever possible to increase reuse possibilities.
- There is a need for a way to handle the modelling of arrays in Asbru., especially for the modelling of tables with more than two dimensions.
- A type declaration should be added to Asbru, including the possibility of sets.
- Variables and parameters should have an initial value, preferably a standard value such as ‘undefined’.
- Add to Asbru a way to assign a return value to a variable.
- Abstractions would be a valuable addition to Asbru, yet should be used with reservation.
- Functions are needed to cope with mathematical operations required to determine certain variable values.
- To check whether or not the intention of a plan is accomplished, a specific modelling strategy is recommended.
- Two different kinds of design patterns are identified. The first would create a sort of jurisprudence, the second concerns the use of subroutines. Both, though especially the first, are recommended.
- A TRY-ALL functionality would be a great addition to the representation of sequences in Asbru.
- Allow the nesting of different plan types within a single plan body.
- Expand the possibilities for using the If-Then-Else statement.
- Didactics and supplemental material are part of guideline and need modelling. This should be added to Asbru.
Observation
- Asbru's ability to model temporal information is very extensive, and sometimes allows more than one way to model the same temporal information.

4.3 Discussion

There were multiple inputs for this study: The guideline, the modelling language and the modeller. Before assigning the results and the conclusion to the modelling language it must be determined to what extent these results are biased by the guideline or the modeller. Whereas the guideline can be biased by not being representative for medical guidelines, the modeller can influence the model on which the results are based or make subjective decisions and interpretations. The determination of the extent of bias in the results of research question one, the

previous aspects are discussed here. After this the lessons learned are briefly discussed.

Representative Guideline?

The main point of discussion for deciding on the validity of the previous results in relation to the posed research question would be the question whether or not these results are valid for medical guidelines in general, or very specific to the guideline used in this study. In paragraph 4.1.2 the hyperbilirubinemia guideline was compared to the five types of guidelines identified by the institute of medicine, and found to be a combination of three different types. This in itself means the studied guideline is not representative for the two other kinds of guidelines, although this doesn't necessarily mean the results are not applicable to these kinds of guidelines.

Based on the formats present in the hyperbilirubinemia guideline (narrative text, lists, tables and flowcharts), the guideline can be considered representative for the majority of guidelines. A more recent development in the field of guidelines is the increasing amount of evidence based guidelines. Examples of such evidence based guidelines are the 'Recommended Breast Cancer Surveillance Guidelines' adopted by the American Society of Clinical Oncology [33]. One of the reasons the hyperbilirubinemia guideline is not considered representative for this kind of guideline is it is not uncommon in evidence based guidelines for recommendations to be accompanied by a 'grade of recommendation', stating the kind of evidence the recommendation is based on and a 'level of evidence', indicating the consistency of the evidence or lack thereof. It would be recommendable to extend the possibilities of Asbru to enable the modelling of this kind of information (providing every plan with a level of evidence indicator, and a grade of recommendation). This would not only enable Asbru to model these evidence based guidelines, it would also provide additive information to critiquing systems or decision support systems. In such a situation it would be preferable when standard definitions for levels of evidence and grades of recommendation are used, or at least comparable definitions are used in all guideline models.

Subjective modelling by the modeller.

Since most results are based on the modelling exercise, one could question whether or not some of these results depend on the modeller instead of the guideline. A GLIF study [37] in which individual physicians and computer scientists had to generate a representation of the same guideline found variability in the construction of these representations, though mostly in the physicians model caused by the physicians understanding of the underlying pathology. Although computer scientists developed more literal representations than physicians, this means the content of the model could be biased by the modeller. Although this means different modellers could create different models it is very likely they will need the same representation language possibilities to model the guideline, and will encounter the same limitations of these possibilities or the way these possibilities have to be used.

Subjective interpretation by the modeller.

Remains the possibility of subjective interpretations on part of the modeller. The theoretical possibilities of the modelling language could be studied fairly objectively. Although the determination of the need of certain possibilities to model the guideline can sometimes be a bit subjective they are always based on guideline content, and

whether or not these possibilities are covered by the modelling language can be determined rather objectively. Whether or not these needed possibilities are practically feasible can sometimes be determined objectively, though other times they are a lot more subjective. Questions on what conditions must be met to make something theoretically possible practically feasible can be determined objectively, for instance the use of variables: when they can be used, they must be declared somewhere, need default values et cetera. The subjective questions on practical feasibility tend to be depending on a personal opinion, 'when does an increased model size start to become bothersome?', or 'when does the modelling effort required to model a certain aspect get too high?'. A lot of this subjectivity has been avoided by rephrasing these questions into related questions that can answered more objectively: 'How can the model size be limited to plans containing relevant, guideline based information', and 'how can the modelling effort be reduced?'. This way the recommendations are based on more objective answers.

Results

Asbru provided little guidance for the modelling process. A manual of the language did not exist so there were no instructions available, and here were no tools to assist in the modelling process. This lack of guidance was severely felt, and was the main motivation for the development of the intermediate template, and the recommendation on the creation of jurisprudence as mentioned in the section on design patterns. The absence of an intermediate model was already detected in the comparison of section 2.5 of this thesis, and is considered a necessary addition to the Asbru language, even more so since other guidance material on the use of the language was unavailable as previously mentioned.

The only recommendation on guideline information that could not be modelled is the addition of didactical and supplemental material. When we take a look at the remainder of the recommendations it is remarkable that a rather significant part of these recommendations is concerned with aspects commonly present in programming languages. Perhaps the development of modelling languages should be viewed more often from a programming point of view.

4.4 Conclusion

The investigation on whether the Asbru language is usable for representing medical guidelines was viewed upon from both a theoretical and a practical point of view. After a conclusion will be formulated on the theoretical part of the question, the practical aspect will be addressed.

Investigation whether the Asbru language is useable for representing medical guidelines, theoretically.

As the discussion pointed out, the results of this study are based on a medical guideline which cannot be considered representative for all medical guidelines. The studied guideline can however be considered representative for certain medical guidelines, as described in 4.1.2, and the conclusion will be stated based only on these kinds of guidelines.

Did Asbru cover all (medical-) aspects encountered in the studied guideline? When the recommendations are viewed in light of this question, the lack of a way to model didactical material is relevant to the complete modelling of guideline content. It is an important aspect as guidelines often include didactic material, even though it is sometimes as little as a prevalence of the disease. The didactical information does however not impact the care strategy presented in the guideline. Still the value of including (supporting-) information should not be underestimated, as it can influence the acceptance of the guideline by the user, and thereby increase guideline compliance. Furthermore including didactical material creates a possibility for the guideline to serve as an educational tool.

The recommendations made under the headings 'datastructures' and 'Functions' did not originate directly from guideline aspects that needed modelling. They are however necessary to allow a meaningful modelling of certain guideline aspects. They are very basic necessities and strongly needed to make the Asbru language serviceable for modelling medical guidelines.

Investigation whether the Asbru language is usable for representing medical guidelines, practically.

For aspects to be practically feasible they first need to be theoretically possible, which means a coding must exist. Besides this the coding should not be extremely long-winded, and the required modelling effort should be reasonable. Discursive modelling was encountered for example under the heading 'representation of sequences'. The recommendations on design patterns are concerned with decreasing the overall modelling effort and ease the modelling process, as does the introduction of an intermediate model. The introduction of these recommended aspects in the Asbru language would definitely reduce modelling effort and modelling size, thus increase the practical feasibility of modelling a guideline using the Asbru approach.

It is concluded that the Asbru 6.5b approach provides a representational language which is reasonably usable for modelling medical guidelines, though a number of recommendations to extend its usability are made. Main weakness of the language is however the lack of support during the modelling process itself. The implementation of the recommendations made in this chapter could enhance Asbru's usability as a modelling language. As some of these recommendations decrease modelling effort and increase support during the modelling process itself, Asbru's feasibility would increase as well. At the time of this writing a number of these recommendations have found their way into Asbru version 7.2. There is still room within the Asbru approach for extensions and improvements, making this a very promising modelling language.

5 Guideline verification

When a paper-based guideline is modelled using a guideline modelling language such as Asbru, certain anomalies of this paper-based guideline might be uncovered. This chapter is dedicated to the question whether or not this is the case, and what kind of anomalies were found. Reduction of anomalies would lead to an increased internal quality of the guideline. The external quality of the guideline can only be increased by panels of domain experts and research, not by formalising it, since a guideline containing wrong recommendations might prove completely free of anomalies.

5.1 Materials and methods

Material

The material used for this study was the hyperbilirubinemia guideline which has been described in paragraph 4.1.2, and the guideline model as present in the appendix. This model is based on the Asbru syntax version 7.2, in which a number of adaptations were made not present in version 6.5b. These adaptations did not concern the kind of information that can be modelled in Asbru but influenced especially modelling technical aspects of Asbru, such as the allowance of multiple plan types within a single plan body and increased allowance of the use of If-Then-else statements. Because of the nature of these adaptations the 7.2 model did not influence this study. Main reason for choosing the 7.2 model was the elimination of a few imperfections still present in the 6.5b model.

Method

The verification process was started by comparing the guideline content to the guideline model. This way one can check whether or not all of the guideline content found its way into the model. Every time information is found in the guideline that's not present in the model there are two possibilities:

1. The model is incomplete, the aspect should have been present.
2. There's a distinct reason for the aspect not to be present in the model.

The first case is actually a verification of the guideline model, which is a nice side-effect. The second case is more interesting for this study, since these distinct reasons are either an imperfection of the Asbru language, irrelevant information, or the anomalies we're trying to find. Besides the comparison of guideline content with the guideline model, the model itself was critically examined. This critical evaluation starts inevitably at the beginning of the modelling exercise as the guideline has to be read carefully and modelling decisions have to be made. This already leads to the detection of some indistinct aspects and inconsistencies. Still another critical evaluation was performed after the model was completed, as not to let any anomalies go unnoticed. Summarising the verification process consisted of three different phases:

1. The modelling exercise
 2. The comparison of guideline content and guideline model.
 3. Critical examination of the guideline model.

5.2 Results

Almost all discovered anomalies in the guideline were found during phase one: the modelling exercise, since modelling decisions had to be made concerning these aspects. The comparison of guideline content and guideline model was especially useful for verification of the model. Found anomalies were already detected during the modelling process. The anomalies found in phase three were mostly already discovered during phase one, but were either remembered again or better identified because of the critical evaluation. The process of formalisation was thus found to be very valuable to the verification of the guideline. The results are presented in an ordering based on the different kind of irregularities that were discovered, being '*Ambiguous guideline parts*', '*Missing information*' and '*inconsistencies*'. The results presented in these sections are a complete listing of the anomalies discovered.

Ambiguous guideline parts

One of the first things one does when initiating a modelling exercise of a guideline is the reading of this guideline. Just by careful reading an ambiguity was detected in point 9 of the guideline. In this case the ambiguity was caused by uncertainty whether or not the first and second sentence should be read separately or in relation to each other. It is an example of an unclear recommendation.

Another case of ambiguity are the comments in respect to term infants that are less than 24 hours old. The algorithm states the process starts with a pediatrician evaluating the term newborn with *jaundice*. Box 16 of the algorithm states when the infant is '*jaundiced and ≤ 24 hours of age*' 'Individualised clinical evaluation is necessary', '*exit algorithm*'. This is in correspondence to the comment near table two, stating clinically jaundiced children of ≤ 24 hours old are not considered healthy, thus this guideline is not applicable to them. At least the above statements seem to be in correspondence with each other, though '*jaundice in box 16*' is suddenly '*clinically jaundiced*' in comment table two. Perhaps '*clinically jaundiced*' is the same as '*clinically significant jaundice by medical judgement*' in box 18 of the algorithm, yet '*clinically significant jaundice by medical judgement*' is only determined in children older than 24 hours since children younger than this were already excluded from the algorithm in box 16, and the comment in table two clearly speaks of ≤ 24 hours. Remains only the possibility that '*clinical jaundice*' is the same as '*jaundice*' in box 16. This would mean children with jaundice, younger than 24 hours of age must exit the algorithm (box 16), since they are not considered healthy (comment table two). This is in correspondence with the title of the guideline stating the guideline is meant for '*Healthy*' newborn. Yet to increase the confusion the guideline does make recommendations on the evaluation and management of children jaundiced in the first 24 hours of life (evaluation number 5, management number 1). Previous example is a very important indistinctness of the guideline as it effects the applicability of the guideline.

Missing information

As the above example shows, terminology is used but not clarified, just like the terms 'jaundiced', 'clinical jaundice', and 'clinically significant jaundice'.

Sometimes non quantitative terms were used in the guideline, which are impossible to model. When 'a rapid increase in TSB-level' (table 1, management number 1) is not accompanied by information on when an increase is considered 'rapid', one can make 'rapid increase' an abstraction, yet is unable to define this abstraction. This missing information is even worse when other actions depend on it. This is the case in paragraph two of the treatment section, which states:

"Determination of the rate of rise of TSB and the infants age may help determine how often to monitor bilirubin levels and whether to begin phototherapy". This is an very vague part of the guideline, as it is not even sure a relation should exist between the rate of rise and how often to monitor TSB levels. And when this relation would exist, there is no information on what this relation would look like: how many times monitoring at what rate of rise of TSB. Whether or not to begin phototherapy would also be depending on this rate of rise of TSB, and again more detailed information is missing.

Another kind of missing information is the unclearness on the reasons for performing certain tests like blood typing in this guideline. The reasons for the maternal blood typing are given in the text (number 1, 2 and 3), although an inconsistency exists between number 2 and the algorithm box 10 on whether or not a mothers negative rhesus factor is an indication for performing a coombs test and blood typing on the infants blood. As the results of the coombs test are used as a condition to exit the algorithm, the results from the infants blood typing are of no importance anywhere in the guideline.

As mentioned in previous chapters, intentions can be modelled in Asbru and can be a very important aspect. During the knowledge acquisition phase an attempt was made to acquire information on intentions, since in most cases this information was not present in the guideline. This is a different kind of guideline incompleteness than the incompleteness described in the last paragraph. Main difference is in the other cases the guideline was the standard and the question was whether or not Asbru could model the guideline. For this aspect suddenly the Asbru possibility of intentions is viewed as the standard, and the question is whether or not the guideline should contain information that can be modelled in Asbru. Therefore the lack of intentions in the guideline can not be considered an internal guideline anomaly, and therefore not missing information but a possible useful addition.

Inconsistencies

An inconsistency concerning the blood typing has already been presented in the previous paragraph since it was connected to an example of missing information. The other inconsistencies are presented here.

The second sentence of paragraph two in the treatment section is a major inconsistency in respect to table two and actually the rest of the treatment section, as it states:

“Continued observation may be an appropriate alternative to repeated TSB testing and phototherapy”. This would mean every time phototherapy is recommended in the section on ‘management of Hyperbilirubinemia .. by age’ or in table two, continued observation could have been an appropriate alternative, though neither text nor table confirms this. When the statement is only meant for the column called ‘Consider Phototherapy’ from Table two, it should have been specified in what cases continued observation might be an appropriate alternative to phototherapy and repeated TSB measurements. Other inconsistencies were the one found between the evaluation section number three and the algorithm, in which different conditions for recommending the saving of cord-blood are given, and between the evaluation section number nine and the algorithm concerning recommended actions after light-stools or dark-urine are found and when jaundice persists beyond three weeks.

An inconsistency, or at least an incompleteness, was found concerning the enumeration of aspects which should be investigated to rule out any underlying illness. The list of aspects described in table one is the more extensive of the two and seems to cover all aspects, though the list in the evaluation section number 7 includes behaviour changes, an aspect not present in table1. Although the list of table one was incomplete compared to the text, it contained a surplus in relation to the algorithm. Three anamnesis questions present in the list were of absolutely no consequence in the algorithm. Apparently the algorithm was more strict since it needed less information to proceed to the next step in the care process. In the model itself these parts of the guideline have been modelled in plan 1.4.1.3. This plan is a compromise as all the questions present in the text and table 1 are asked, yet only a few of these questions have an informational value, like it is written in the algorithm. During the careful evaluation this guideline inconsistency became apparent again, since three variables remained unused, a strategy chosen during the modelling exercise in which the algorithm prevailed over the list of table one.

5.3 Discussion

Perhaps the clearest way to discuss the results of the previous paragraph is to discuss the sections on ‘Ambiguous guideline parts’, ‘Missing information’ and ‘Inconsistencies’ separately. Each raises its own questions, and only one question is the same in all three sections: *“Does it help to formalise the guideline to discover these irregularities?”*. This question is discussed for all three sections together in the final paragraph.

‘Ambiguous guideline parts’

The results section on ambiguous guideline parts consisted of two examples. As ambiguity is sometimes caused by misunderstanding, why are these two examples indicate a mistake in the guideline instead of a misunderstanding on the reader’s part. The first example had to do with the question whether or not a previous sentence described the context of the following sentence. This is actually more of a linguistic issue. Take for example the sentence ‘Tomorrow we’re visiting my auntie’, followed by the sentence ‘I’m not feeling very well’. Both sentences could very well have nothing to do with each other, but they might. This could mean I am not feeling very

well *because* we're going to visit my auntie, an example in which a previous sentence provides the context for the next. Maybe I'm not feeling very well *despite* the fact we're going to visit my auntie, in which case the previous sentence again provides a context for the second one, yet my relation with my auntie has changed dramatically. Or maybe I am never feeling well the day before we're going to visit my auntie, in which case my auntie has nothing to do with it and the context has changed to time. Although this example might be a bit childish, it does show very well the ambiguity that arises when a previous sentence would implicitly provide the context for the following one. Therefore sentences should always be read separately from each other, unless an explicit reference is present. They should never be written in a way which implies the possibility of a relationship between the sentences. This implication is unfortunately present in the guideline, as they share the same reference number and a certain chronology is implied. In the second example the ambiguity finds its root in an inconsistent use of terminology (jaundiced, clinically jaundiced and clinically significant jaundice), terms that are used yet not clarified. It's just this clarification which might have saved the writers from their inconsistency. The combination of the inconsistent use of terms, and the missing information which should have clarified these terms is the main cause for the ambiguity on whether the guideline is appropriate for neonates of age less than 24 hours. This is considered a very important guideline error, since it should always be clear to whom the guideline is applicable.

Missing information.

The missing information concerned a number of terms such as clinically jaundiced. These terms might be very clear to the guidelines target group and therefore omitted from the guideline, although this target group is rather heterogeneous. This question was not posed to people from the target group, making it unknown whether or not this should correctly be considered 'missing information'. The same applies for the missing information on tests like the children's blood typing. Domain experts might know exactly why this test should be done, and what its results imply. Still this kind of information is very different from the previously mentioned missing definitions since this blood typing test is part of the care process involved in case of jaundice. It should have been present in the guideline, if only to create a complete guideline without any 'lose ends', no matter whether or not this knowledge is commonly present among domain experts.

Last aspect of the missing information section was the incomplete presence of explicit intentions. As already mentioned in the results, this information is lacking from a guideline modellers point-of-view who's working with a modelling language which allows the representation of intentions. While working with our domain expert it became clear the intentions behind actions were no mystery to her, and therefore the unavailability of some explicit intentions can not be considered a guideline anomaly. It's a modellers wish to make things a little easier, as intentions can be very useful in for example critiquing systems.

Inconsistencies.

Most inconsistencies were differences between the text of the guideline and the algorithm. An algorithm could be a very helpful addition to a guideline as it provides easy overview of the ordering of actions. It should however be consistent with the

recommendations made in the text of the guideline as it should be based on this text. An algorithm which differs in recommendations from the text is a new guideline in itself. Other inconsistencies arose between lists, tables and text. Although it can be very helpful to provide information in a different format, the information itself should be the same. These inconsistencies are a bit sloppy on part of the writers of the guideline, and could have been avoided. Some inconsistencies undermined or questioned the guidelines own recommendations. A little sentence such as 'observation might be an appropriate alternative to repeated TSB measurements and phototherapy' has major implications. It seems as though the writers were afraid of making real recommendations and decided to give the appearance of recommendations, yet concurrently undermining the value of these recommendations. As they mentioned in the introduction to the guideline they attempted to describe a range of acceptable practices as no scientific evidence is available. If this is how they would have made the recommendations it wouldn't have been a problem, yet they describe rather detailed the management of hyperbilirubinemia including specific TSB boundaries, without mentioning appropriate alternatives as identified by the committee of writers. Their alternative is the one sentence as mentioned previously. The vagueness of the guidelines structure, and the vagueness or inconsistency of the recommendations in the guideline create a bad atmosphere for the transaction of knowledge.

"Does it help to formalise the guideline to discover these irregularities?"

This is the main question which has to be answered. Unfortunately it's also the most difficult to answer, since it can not be compared to guideline verification done without the use of a formalising route. Formalisation does however provide a good foundation for detection of irregularities, as it obliges a critical examination of the guideline. Formalisation requires rather black and white guideline information. Everything within the greyish area needs more careful investigation as it raises questions on how to model it. Probably a number of irregularities would also have been detected by careful reading and dissemination of the guideline information, though one can assume this has been done as the hyperbilirubinemia guideline is included in the *National Guideline Clearinghouse*, and is issued by the American Association of Pediatrics as a standard. Formalisation does help to discover irregularities in a guideline, though it is a weigh up between a more complete detection of irregularities by formalising which is quite a lot of work, and a less complete detection which would likely take less effort.

5.4 Conclusion

A significant number of flaws were detected in the paper based original guideline by formalising it. Although as mentioned in the discussion it cannot be proven these flaws would not be detected by careful reading and guideline dissemination, this is of minor importance. Fact is the formalisation route provided a sound foundation for verifying the guideline as it required a detailed examination of every aspect, and is thus helpful to the discovery of guideline irregularities. It would be interesting to investigate and compare the results of a more mathematical approach to the verification of this guideline with the results obtained in this study.

6 Reflection

Though the Asbru language is far from perfect (especially version 6.5b) and a significant number of recommendations are made, it is a promising language. There are enough possibilities to increase both its usability for modelling guidelines, not just medical ones, and its practical feasibility. The development of user friendly tools to aid in the modelling process would make the Asbru language more open to the public, increasing its use by different modellers and thereby contribute to the development of the language itself by increasing both usability as practical feasibility. The great expressive possibilities the Asbru language already has contribute to its use for the verification of the guideline by formalisation. Formalisation proved a useful technique for the detection of guideline irregularities, and it provided a solid foundation for this detection. This verification of a guideline could also be done in a more formal way by using mathematical techniques, a very interesting subject for future research. Work on this is being done by the KIV project of the university of Augsburg [38].

As mentioned in the introduction, medical guidelines are becoming increasingly important in today's healthcare. Yet so many guidelines are developed, it's becoming near impossible for physicians to read them all and stay up to date with all the adaptations, let alone use these guidelines as reference books in the daily care process. Computerised support of guidelines could greatly enhance the use of guidelines, and even ease updating of guidelines by the automatic downloading of newer releases, thereby ensuring the physicians have access to the latest available information. Languages like Asbru that allow the translation of paper based guidelines to versions that are readable and interpretable by a computer are therefore very important developments. As these languages get better and better, and close to fulfilling their role in the enhancement of the use of guidelines, do guidelines get better and better as well? As shown in chapter five of this study there is definitely room for improvement, as it is unlikely the guideline used in this study is the only one with flaws. A high guideline quality is however more than the absence of ambiguities, inconsistencies and missing information. High guideline quality also means the correct recommendations are made, preferably based on sound research and as recent as possible. In our point of view evidence based guidelines will therefore become even more than today a necessity in healthcare. Especially guidelines based on sound scientific evidence, with clear recommendations and a high internal quality should be translated for computerised support. Maybe even more important than the question whether guideline modelling languages are good enough to model guidelines, is the question whether guidelines are good enough to be modelled. The paradox is that guideline quality can improve by formalising it, yet a certain guideline quality is needed before the guideline should be formalised to use for computerised support. Perhaps the formalisation process should be performed twice, with different goals in mind: First a formalisation which aims at increasing the guideline quality followed by a discussion with medical professionals, and a second one aiming at achieving a correct representation of this 'enhanced' guideline.

Once guidelines achieve such a quality level, all these guidelines need modelling. The modelling of all existing and future guidelines would be an incredible task, which is why the modelling effort needed to model a single guideline should be limited as much as possible. This effort can be decreased in a number of different ways. One of these is the reuse of parts of an existing guideline model. An Asbru plan might be reusable, yet this will not often be the case as it contains a lot of information, like conditions, that must be exactly the same in the other guideline to make it reusable. If we divide a plan into a plan body and the rest, the plan body might be reusable more often. The less information a guideline part contains, the more easier it can be reused, yet the less it decreases the modelling effort. A plan which content can be varied to make certain adaptations, whereas the rest of the plan can be reused could provide a solution. Much like this is the development of both types of design patterns as described in this study. Design patterns can be used to reuse parts within one guideline like subroutines, or for cross-guideline reuse. Cross-guideline reuse can be problematic, as an update of guideline X might influence all guidelines that reuse parts of guideline X. For this reason it's probably best to concentrate on reuse possibilities within a single guideline model, and design patterns which provide preferred modelling strategies in standard modelling situation. Design patterns as examples of preferred modelling strategies could aid in the creation of good quality models that are relatively easy to make. Another way to decrease the modelling effort is to provide tools to ease the modelling process. Besides decreasing the effort this would probably increase the model's quality since the easier it is, the less mistakes are made in the translation process. It would after all be a shame when a high quality paper based guideline would result in a poor quality electronic version. Besides, the easier it is, the more it is likely that one day guidelines will immediately be developed in an electronic format.

The development of good guidelines and the development of good guideline modelling languages and tools have a lot to benefit from each other, and should not be done separately from one another. It's a great example of an area in which the medical information scientist can participate as an intermediary between the physicians, who need to develop a guideline suitable for modelling, and the developers of the modelling language, who need to develop a language suitable to the medical domain. Also in modelling process itself the medical information scientist provides useful contributions, bridging the gap between the medical knowledge, and the technical aspects of modelling formalisms.

7 Bibliography

- [1] Plato. *Statesman*. Annas J, Waterfield R, eds. Cambridge: Cambridge University Press, 1995 xvi-xvii, 60-61.
- [2] Hurwitz B. *Legal and political considerations of clinical practice guidelines*. BMJ 1999; 318: 661-664.
- [3] Woolf S.H., Grol R., Hutchinson A., Eccles M., Grimshaw J. *Potential benefits, limitations, and harms of clinical guidelines*. BMJ 1999; 318: 527-530.
- [4] Woolf S.H. *Practice Guidelines, a new reality in medicine*. Arch Intern Med 1992; 152: 946-952.
- [5] Clayton PD, Hripcsak G. *Decision support in healthcare*. Int. J. of Bio-Medical Computing, 1995; 39: 59-66]
- [6] Grimshaw J.M., Russell I.T. *Effect of clinical guidelines on medical practice: a systematic review of rigorous evaluations*. Lancet 1993; 342: 1317-1322.
- [7] Miksch, S., Shahar, Y. & Johnson, P. *Asbru: A Task-Specific, Intention-Based, and Time-Oriented Language for Representing Skeletal Plans*. 7th Workshop on Knowledge Engineering: Methods & Languages (KEML-97), Milton Keynes, UK, 1997; January; 22-24.
- [8] Ohno-Machado L., Gennari J.H., Murphy S.N., Jain N.L., Tu S.W., Oliver D.E., Pattison-Gordon E., Greenes R.A., Shortliffe E.H., Octo Barnett G. *The Guideline Interchange Format: a model for representing guidelines*. JAMIA 1998; 5: 357-372.
- [9] Fox J., Johns N., Lyons C., Rahmanzadeh A., Thomson R., Wilson P. *PROforma: a general technology for clinical decision support systems*. Comp. Methods and Programs in Biomedicine, 1997; 54: 59-67.
- [10] Pattison-Gordon E., Cimino J.J., Hripcsak G., Tu S.W., Gennari J.H., Jain N.L., Greenes R.A., *Requirements of a sharable Guideline Representation for Computer Applications*. Stanford University, Report No. SMI-96-0628, 1996.
- [11] Field M., Lohr K.(eds), *Clinical Practice guidelines: Directions for a New Program*, Institute of Medicine, Washington D.C.,National Academy Press, 1990.
- [12] Miksch S., *Plan management in the medical domain*. AI communications 1999; 12; 209-235

- [13] Field M.J., Lohr K.N.(eds), *Guidelines for clinical practice: From development to Use*. Institute of Medicine, Washington D.C.,National Academy Press, 1992.
- [14] Pattison-Gordon E., *ODIF: Object Data Interchange Format*. Decision Systems Group, Brigham & Woman's Hospital, Boston Massachusetts 02115, Technical Report Number: DSG-96-04; 1996.
- [15] Introduction to the Intermed common guideline model, and the Guideline Interchange Format. Available at:
http://dsg.harvard.edu/public/intermed/glif_overview.html.
- [16] Duftschmid G., *Knowledge-based Verification of Clinical Guidelines by Detection of Anomalies, Dissertation*. 1999.
- [17] Musen M.A., Tu S.W., Das A.K., Shahar Y., *A component based architecture for automation of Protocol-Directed Therapy*, in: Barahona P., Stefanelli M., Wyatt J. (Eds), Proc. 5th Conf. Artif. Intell. Europe, Pavia, Italy. 1995; 3-13
- [18] InferMed Ltd., Arezzo. <http://www.infermed.co.uk/>.
- [19] Shahar Y., Miksch S., Johnson P., *The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines*, Artificial Intelligence in Medicine 1998; 14; 29-51
- [20] Hripcsak G., Ludemann P., Pruor T. A., Wigertz O. B., Clayton P. D. *Rationale for the Arden Syntax Computers in Biomedical Research* 1994; 27; 291-324.
- [21] Miksch S., Kosara R., Shahar Y., Johnson P., *AsbruView: visualisation of time-oriented skeletal plans*. Proceedings of the fourth International Conference on Artificial Intelligence Planning Systems 1998 (AIPS-98). Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, AAAI Press 1998; 11-18.
- [22] Shiffman R. N., Brandt C.A., Liaw Y., Corb G. J., *A Design Model for Computer-based Guideline Implementation Based on Information Management Services*. JAMIA 1999; 6: 99-103.
- [23] Friedland P. E., Iwasaki Y., *The Concept and Implementation of Skeletal Plans*. Journal of Automated Reasoning 1985; 1; 161-208.
- [24] University of Heidelberg, Department of Medical Informatics. *The Sisyphus IV project*.
http://www.med.uni-heidelberg.de/mi/projects/sisyphus/sisyphus_IV.html

- [25] Practice guideline of the American Academy of Pediatrics (AAP), *Management of hyperbilirubinemia in the healthy term newborn*. Pediatrics 1994; volume 94(4).
- [26] Heese H. v/d, *Handbook of pediatrics*. Oxford university press. 4th edition, 1995; 226-9, 356-63.
- [27] J.G.Koppe e.a. *Neonatologie*,. Bohn, Stafleu, van Loghum. 1992; 2de druk; 34-9.
- [28] Taelisch H.W., Ballard R.A, *Avery's diseases of the newborn*. 7th edition; 1998; 995-1001.
- [29] Fetter W.P.F. et. al. *Hyperbilirubinemia: richtlijnen voor diagnostiek en behandeling*. *Nederlands Tijdschrift voor de Geneeskunde*. 1997;141-3
- [30] Hart A., *Knowledge elicitation: issues and methods*. Computer aided design.1985; 17; 455-462
- [31] Geiwitz J., Kornell J., McCloskey B, *An expert system for the selection of knowledge acquisition techniques*. Technical report 785-2, contract no. DAAB07-89-C-A044. Anacapa sciences, 1990.
- [32] Cordingley E.S., *Knowledge elicitation techniques for knowledge-based systems*. In: Diaper D. (Eds), *Knowledge elicitation: Principles, techniques and applications*. Ellis Horwood Ltd. 1989.
- [33] American Society of Clinical Oncology, *Recommended Breast Cancer Surveillance Guidelines*. *Journal of Clinical Oncology*, 1997; 15; 2149-2156
- [34] AMC richtlijnen voor postoperatieve behandeling Hartpatienten jan 1995 versie 1.1.
- [35] Shiffman R.N., Brandt C.A., Liaw Y., Corb G.J., *A Design Model for Computer-based Guideline Implementation Based on Information Management Services*. *JAMIA* 1999; 6; 99-103.
- [36] Teach R.L., Shortliffe E.H., *An analysis of physician attitudes regarding computer-based clinical consultation systems*. *Computational Biomedical Research* 1982; 14; 542-58.
- [37] Patel V. L., Allen V.G., Arocha J.F., Shortliffe E.H., *Representing Clinical Guidelines in GLIF: Individual and Collaborative Expertise*.
- [38] KIV

8 Appendix

Appendix A : Guideline '*Management of hyperbilirubinemia in the healthy term newborn*'.

Appendix B : Guideline Model, Asbru version 7.2