

Report on modelling medical protocols in Asbru (illustrated with jaundice protocol)

Mar Marcos

Plan

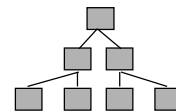
- Introduction
- The Asbru language
- The jaundice protocol
- Modelling jaundice protocol in Asbru
- Lessons learned

Introduction

- Purpose:
 - describe our experiences in modelling protocols in Asbru
- Motivation:
 - it is a difficult task
 - lessons learned can be useful

The Asbru language

- AI language intended to support protocol-based care tasks
- Some features:
 - protocols are described as hierarchical, skeletal plans
 - plans can be executed, e.g. sequentially or in parallel
 - conditions can be defined to control plan execution, e.g. filter preconditions (applicability)
 - intentions can be stated for plans
 - conditions and intentions can have time annotations, e.g. between 2 and 4h after certain event



The jaundice protocol

- The jaundice protocol of the AAP*:
 - management of jaundice in healthy term newborns
 - aspects of clinical practice: diagnosis & treatment
 - features: short time-span application, non-specialist/hospital use, low-frequency data
 - presentation: 8 pages with text + lists + tables + flowcharts

* Included in <http://www.guideline.gov/>

Modelling jaundice protocol in Asbru

- First attack: putting the protocol into Asbru
 - familiarisation with the protocol
 - analysis of the protocol
 - modelling of the protocol: top-down, iterative refinement (in plans and details thereof)
 - first protocol model in an intermediate Asbru notation (**intermediate model**)
- Observation:
 - intermediate model bigger than the original protocol

Modelling jaundice protocol in Asbru (cont.)

– intermediate Asbru notation:

Hierarchical number	Plan name	
1	PLAN Regular-treatments	Hyperbilirubinemia
1.1	PLAN BODY	Check-for-rapid-TSB-increase
1.2	PLAN 1.5.1	Regular-treatments
1.3	DIAGNOSTIC PLANS	do any-order, wait-for all
1.4		AVOID INTERMEDIATE STATE:
1.4.1		bilirubin = transfusion
1.4.2		ACHIEVE OVERALL STATE:
1.4.2.1		bilirubin = observation
1.4.2.2		do any-order, retry-aborted-subplans=yes, wait-for Observation
1.4.3		CONDITIONS: FILTER (bilirubin not transfusion * [_,][0,][_,] *now*)
1.4.3.1		ABORT: (bilirubin transfusion * [_,][0,][_,] *now*) ...
1.4.4	PLAN BODY	Phototherapy-intensive
1.5	TREATMENT PLANS	Phototherapy-normal-prescription
1.5.1		Observation
1.5.1.1		DO ALL UNORDERED {
1.5.1.2		Feeding-alternatives
1.5.1.3		DO-SOME-ANYORDER {
1.5.1.4		Retry aborted children
1.5.1.5		Continuation specification: (Observation)
1.5.2		Phototherapy-intensive
		Phototherapy-normal-prescription
		Phototherapy-normal-recommended
		Observation } }

Modelling jaundice protocol in Asbru (cont.)

- Next step: putting the intermediate model into “real” Asbru
 - using Asbru DTD + professional XML editor
 - protocol model in Asbru syntax (**XML-Asbru model**)
- Observations:
 - increasing complexity and size (~70K) and decreasing understanding/readability
 - tool support becomes a real need!

Modelling jaundice protocol in Asbru (cont.)

– Asbru syntax:

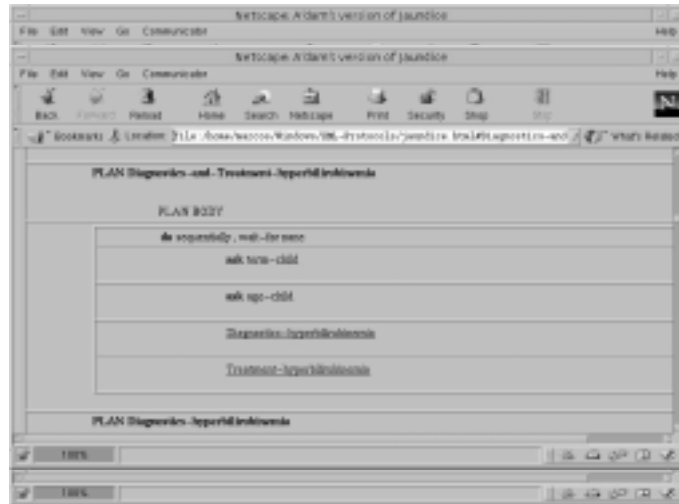
```
<plan name="Regular-treatments">
  <intentions>... </intentions>
  <conditions>... </conditions>
  <plan-body>
    <subplans type="unordered">
      <wait-for> <all/>      </wait-for>
      <plan-activation>
        <plan-schema name="Feeding-alternatives"/>
      </plan-activation>
      <subplans type="any-order" retry-aborted-subplans="yes">
        <wait-for>
          <static-plan-pointer plan-name="Observation"/>
        </wait-for>
        <plan-activation>
          <plan-schema name="Phototherapy-intensive"/>
        </plan-activation>
      </subplans>
    </subplans>
  </plan-body>
</plan>
```

Modelling jaundice protocol in Asbru (cont.)

- Last step: validating the XML-Asbru model
 - with the help of specifically developed tools:
 - visualisator of XML-Asbru protocols
 - interpreter of XML-Asbru protocols
 - improved protocol model (**final XML-Asbru model**)

Modelling jaundice protocol in Asbru (cont.)

- the visualiser: translates XML protocols into HTML



Modelling jaundice protocol in Asbru (cont.)

- ...allowing an easy reading and navigation of protocols
- the interpreter: reads XML protocols and simulates their execution, allowing the validation of protocol behaviour

(see Tibor's presentation)

Lessons learned

- One-step modelling is not possible, instead:
protocol —→ intermediate —→ XML-Asbru
① model ② model
- Hint for step ①: using protocol elements (e.g. flowcharts) is a good strategy
- Conclusions:
 - protocols rapidly increase in size
 - time required highly depends on modeller's training (on both the protocol and Asbru)

Lessons learned (cont.)

- Conclusions from step ②:
 - protocols increase in size even more
 - addition of details implies a revision of the protocol
 - requires deeper knowledge (on both the protocol and Asbru) and tool support

Final Asbru model of jaundice

...as shown by the visualisator 