

The formalisation of medical protocols: easier said than done

Marije Geldof
7 November 2002



Overview

- Introduction
- Guideline Markup Tool (GMT)
- Project goals:
 - linking with GMT
 - evaluating GMT
 - complexity of Asbru protocols
 - patterns
- Summary
- Future work



2

Guideline Markup Tool (GMT)

- Editor that helps translating guidelines from text into Asbru written in Java
- Developed by Peter Votruba (Vienna)
- Supports the following tasks:
 - authoring guidelines
 - understanding guidelines
 - structuring Asbru (with macro's)



3

Project goals

- Defining the relations between original protocols and their Asbru translation using the Guideline Markup Tool (GMT)
- Evaluating the Guideline Markup Tool
- Determining where the increased complexity of the Asbru protocols comes from
- Identifying patterns in Asbru protocols



4

Linking with GMT (1)

- Purpose:
 - insight in relation between protocol and its formalisation
 - uncovering the reason for increased complexity of Asbru protocols
 - enabling discussion with domain experts
- Linking asks for critical insight in the protocols
- Links can be defined in different ways



5

Linking with GMT (2)

- Results:
 - jaundice: 239 links
 - diabetes: 249 links
 - explicit and implicit relations
 - already uncovered anomalies visualized
 - unmodeled parts of the original protocol visualized
 - new anomalies uncovered
 - unlinked XML parts uncovered



6

Evaluation of GMT (1)

- Current status: important features implemented, user convenience can be improved
- Added functionality:
 - multiple links to XML
 - scrolling through multiple links
 - visualization of linked and unlinked XML parts

Evaluation of GMT (2)

- Missing functionality:
 - linking to pictures in HTML
 - multiple links to HTML
 - composed links
- Desired functionality:
 - editing in HTML view
 - find function in XML view

Complexity of Asbru protocols (1)

- Implicit knowledge
- Plan body type, wait-for-optional-subplans, do type
- Additional knowledge of domain experts
- Retry delay of cyclical plans
- Intentions
- Conditions

Retry delay of cyclical plans

cyclical-plan

evening-dose ← Test-glucose-and-adapt-evening-insulin(evening-dose)

retry-delay: min = 7d, max = 10d

(From Find-evening-insulin-dose)

Intentions

intentions

avoid-non-desired-situations:

avoid intermediate state:

(or (hypoglycemic-coma = true)
(complications = true))

(From Treatments-and-Controls)

Conditions

conditions

filter-precondition: (and (age > 45)
(triennial-visit =true))
(From Anamnesis-older-than-45)

conditions

complete-condition: (alive = false)
(From Treatments-and-Controls)

Complexity of Asbru protocols (2)

- Manually vs. automatically activated plans
- Conversion of parameters to variables
- Prescription of medicine
- Similar plans

Manually vs. automatically activated

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

Conversion of parameters to variables

```

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

    (from Anamnesis-hemolytic-disease)
    
```

Prescription of medicine

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

Similar plans

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

Patterns (1)

- Two kind of patterns
 - in Asbru
 - in medical protocols
- Enables reuse during formalisation
- Patterns can be composed of smaller subpatterns
- Two protocols limited resource for pattern identification

Patterns (2)

```

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

conditions
  activate-mode: manual
  
```

 Protocure

19

Patterns (3)

```

plan [plan-name]
  [comment]
  plan-body user-performed

plan Exit-possibility-of-[x]
  Exit this protocol. Perform appropriate
  laboratory assessment ([comment])
  plan-body user-performed
  
```

 Protocure

20

Patterns (4)

```

plan [plan name]
  [comment]
  plan-body type = sequentially
  wait-for all
    ask [parameter]
    |
    ask [parameter]
  
```

 Protocure

21

Patterns (5)

```

plan [plan-name]
  plan-body type = any-order
  wait-for one
    [plan-name]
    |
    [plan-name]
  
```

 Protocure

22

Patterns (6)

```

intentions
  achieve overall-state: is-known-variable(x)

  later on.....
  if [condition]
  then [x] ← true
  else [x] ← false
  
```

 Protocure

23

Patterns (7)

```

abort-condition: ([x] = true)
  | in plan body
  ask [x]-over
  
```

 Protocure

24

Summary

- Linking gives insight in the relation between the original protocol and its formalisation
- Several aspects that increase the complexity of an Asbru protocol can be identified
- Some patterns can be identified, but two protocols is a limited resource

Future work

- Further development of the Guideline Markup Tool
- Formalisation of protocols using GMT
- Resolving complexity issues
- Identifying more Asbru patterns
- Identifying patterns in medical protocols