

Scriptie

Modelleren van medische richtlijnen

Een vergelijkend onderzoek over PRO*forma* en Asbru

Ruben Meinders

INF/SCR-01-02

Inhoudsopgave

1	Inleiding	3
2	Richtlijnen en representatie van richtlijnen	5
2.1	Medische richtlijnen	5
2.2	Kennistypen in medische richtlijnen	5
2.3	Representatie van richtlijnen: modelleertalen	7
3	Beschrijving van PROforma	9
3.1	PROforma ontologie	9
3.2	Syntax en semantiek van PROforma	10
3.2.1	Representatie van taken	12
3.2.2	Representatie van data-items	13
3.2.3	Expressies in attributen	14
3.3	Uitvoering van een protocol	15
3.4	Softwaretools voor PROforma : Arezzo	16
4	Beschrijving van Asbru	18
4.1	Asbru als onderdeel van het Asgaard-project	18
4.2	De semantiek van Asbru-elementen	18
4.2.1	Plan	19
4.2.2	Time-annotations	21
4.2.3	Temporal patterns (condities)	21
4.2.4	Parameters en variabelen	22
4.2.5	Context	23
4.3	Organisatie en syntax van Asbru	23
5	De cholesterolrichtlijn	25
5.1	De cholesterolrichtlijn: een introductie	25
5.2	Kennisbronnen	26
5.2.1	CBO-consensus	26
5.2.2	Samenwerking met het UMCU	27
5.3	Analyse van de richtlijn	27
6	Modelleren van de cholesterolrichtlijn in PROforma	29
6.1	Representatie van de kennistypen in medische richtlijnen in PROforma	29
6.2	Het PROforma-model van de cholesterolrichtlijn	31
6.3	Bevindingen tijdens het modelleren in PROforma	34

7	Modelleren van de cholesterolrichtlijn in Asbru	37
7.1	Representatie van de informatietypen in medische richtlijnen in Asbru	37
7.2	Het Asbru-model van de cholesterolrichtlijn	39
7.3	Bevindingen tijdens het modelleren in Asbru	42
8	Evaluatie	44
8.1	Uitdrukkingskracht	44
8.2	Begrijpelijkheid	46
8.3	Hergebruik	49
8.4	Gebruik van <i>PROforma</i> en Asbru	50
8.5	Discussie	50
9	Conclusie	52
A	Analyse van de cholesterolrichtlijn	53
A.1	Toegangscriteria voor de richtlijn	53
A.2	Lichamelijk onderzoek	53
A.3	Resultaten uit het lichamelijk onderzoek	54
A.4	Bepaling van het 10-jaarsrisico	55
A.5	Beslissing wanneer statines voor te schrijven	56
A.6	Behandeling met statines	57
A.7	Opmerkingen	57
	Bibliografie	58

Hoofdstuk 1

Inleiding

Ter verbetering van de kwaliteit van medische handelingen, zoals het screenen van patiënten, het stellen van een diagnose en het managen van een behandeling, worden in de gezondheidszorg in toenemende mate richtlijnen gehanteerd ([9, 11]). Deze medische richtlijnen moeten worden gezien als een verzameling uit te voeren acties en bijbehorende condities die aangeven wanneer deze acties moeten worden uitgevoerd ([5, 11]). Het gebruik van richtlijnen boven het ad hoc handelen heeft verschillende voordelen, waaronder kostenbesparing en kwaliteitsverbetering (zie ook hoofdstuk 2).

Het formaat waarin richtlijnen in de regel worden gerepresenteerd, zoals tekst, tabellen, enz., maakt het moeilijk om richtlijnen om te zetten voor gebruik door beslissings-ondersteunende systemen. Daarnaast zijn richtlijnen die in zo'n ad hoc formaat gerepresenteerd zijn, vaak dubbelzinnig en onvolledig. Met het gebruik van zgn. *modelleertalen* voor medische richtlijnen kunnen deze problemen verholpen worden. Zulke talen hebben de eigenschap dat de typen kennis die doorgaans in een medische richtlijn voorkomen, zoals medische handelingen en beslissingen, gemodelleerd kunnen worden in een zodanig formeel formaat zodat interpretatie door een computer mogelijk wordt. Een belangrijk aspect daarnaast is dat bij het modelleren van richtlijnen in deze talen eventueel dubbelzinnigheden en onvolledigheden aan het licht komen.

Deze scriptie beschrijft een onderzoek waarin twee veelbelovende representatietalen, nl. *PROforma* en *Asbru*, vergeleken werden. Voor dit onderzoek werd de cholesterolrichtlijn, zoals deze gebruikt wordt door specialisten aan het Universitair Medisch Centrum Utrecht (UMCU), gemodelleerd in beide talen. Hierbij werd er ondersteuning verleend door experts van het UMCU. Om een beter inzicht te verkrijgen in beide talen werd bovendien gekeken hoe de in medische richtlijnen veel voorkomende kennistypen (met inbegrip van enkele kennistypen die niet in de cholesterolrichtlijn voorkomen) uitgedrukt kunnen worden in *PROforma* en *Asbru*. Kort samengevat kent het onderzoek dus twee doelen:

1. Het modelleren van de cholesterolrichtlijn in *PROforma* en *Asbru*
2. Het vergelijken van de talen aan de hand van een analyse van beide talen en de resultaten uit het modelleerproces.

In het volgende hoofdstuk zal een algemene beschrijving van richtlijnen worden gegeven met daarbij kennistypen die in richtlijnen voorkomen, alsmede het belang van formalisatie van richtlijnen. In de hoofdstukken 3 en 4 zullen resp. de talen *PROforma* en *Asbru* worden uitgelegd. De cholesterolrichtlijn zal in hoofdstuk 5 worden beschreven. In hoofdstuk 6 en 7 zullen voor resp. *PROforma* en *Asbru* de resultaten van het modelleren van de cholesterol-

richtlijn besproken worden. Bovendien zal er in deze hoofdstukken voor beide talen worden gekeken naar de representatie van de kennistypen uit hoofdstuk 2. In hoofdstuk 8 zullen de talen geëvalueerd worden aan de hand van een vergelijking tussen beide talen, en zullen er aanbevelingen worden gedaan. In hoofdstuk 9 volgt tenslotte de conclusie.

Hoofdstuk 2

Richtlijnen en representatie van richtlijnen

2.1 Medische richtlijnen

Bij het uitvoeren van medische handelingen wordt er veelal volgens bepaalde richtlijnen gewerkt. Deze medische richtlijnen beschrijven een gesystematiseerde volgorde van te nemen beslissingen en te volgen stappen, met het doel een bepaalde klinische toestand te bereiken of te handhaven. Er zijn verschillende redenen voor het gebruik van medische richtlijnen:

- Het kan leiden tot het efficiënter inroosteren van personeel en benodigde apparatuur, omdat van te voren beter in te schatten is wat er nodig is voor het verrichten van medische handelingen.
- Aangenomen dat een richtlijn de beste handelswijze representeert kan het leiden tot kwaliteitsverbetering.
- Het kan worden gebruikt voor medisch onderwijs.
- Het kan de arts ondersteunen bij het nemen van beslissingen.
- Op deze manier is het mogelijk om de handelingen van een arts te evalueren.
- De handelswijze zal minder variëren wanneer de behandeling van een patiënt wordt overgenomen door een andere arts.

2.2 Kennistypen in medische richtlijnen

Om een goed beeld te krijgen van medische richtlijnen is het zinvol om te kijken welke typen kennis er zoal in richtlijnen kunnen voorkomen. In diverse artikelen ([5, 9, 10, 11]) zijn reeds diverse eisen aan de uitdrukingskracht van modelleertalen voor medische richtlijnen geformuleerd, hiermee impliciet een beschrijving gevende van de typen kennis in medische richtlijnen. Deze eisen zijn doorgaans geformuleerd vanuit oogpunt van de informatie-technologie. Het beschrijven van deze eisen in medische terminologie leverde onderstaande kennistypen op. Behalve dat deze een beeld geven van medische richtlijnen werden deze kennistypen ook als leidraad gebruikt bij de analyse van de modelleertalen.

Doel

In een richtlijn is meestal, zij het impliciet, het doel beschreven. Soms wordt dit ook voor afzonderlijke stappen binnen de richtlijn gedaan.

Atomaire stap

Hiermee worden stappen bedoeld die niet uit andere deelstappen bestaan. Dit zijn handelingen zoals het meten van de bloeddruk of het opvragen van patiëntgegevens.

Decompositie van stappen

In een complexe richtlijn wordt vaak een groep stappen beschreven als deel uitmakende van één overkoepelende stap. Voor het op deze manier hiërarchisch opdelen van een stap in deelstappen zijn drie redenen. Ten eerste kan de stap op deze manier beter in detail worden beschreven. Ten tweede kan zo worden uitgedrukt dat de deelstappen in een zelfde tijdsbestek moeten worden uitgevoerd. Een derde reden is om hiermee aan te geven dat de deelstappen logisch bij elkaar horen, d.w.z. dat ze bijvoorbeeld van dezelfde ‘soort’ zijn of dat de deelstappen alle door een externe instantie (zoals een laboratorium) worden uitgevoerd.

Procedurele kennis

De volgorde waarin de stappen moeten worden uitgevoerd kan worden gezien als een soort algoritme en is soms grafisch weergegeven, als een soort stroomdiagram. Stappen kunnen *sequentieel* worden uitgevoerd, dat wil zeggen dat een bepaalde stap start wanneer een andere stap is beëindigd. Voor een groep stappen kan worden geëist dat ze tegelijk starten (*parallel*). Hierbij kan het voorkomen dat parallelle stappen *synchron* worden uitgevoerd, dat wil zeggen dat er gewacht wordt totdat al deze stappen beëindigd zijn. Daarnaast zijn er stappen waarvan geëist wordt dat ze *tegelijk moeten eindigen*; het betreft hier typisch taken die zich in de tijd uitstrekken. Denk hier bijvoorbeeld aan het toedienen van een cocktail van medicijnen. Verder kan het voorkomen dat een stap herhaaldelijk wordt uitgevoerd (*cyclische* stappen). Tevens kan een stap zich op een bepaald moment *vertakken*, waarbij de beslissing welke tak wordt gevolgd gebaseerd is op een conditie. Soms komen *optionele* taken voor: het uitvoeren van zo’n stap is niet verplicht. Het wordt dan vaak aan de arts overgelaten of deze de stap wel of niet overslaat.

Precondities

Sommige stappen in een richtlijn mogen alleen onder bepaalde voorwaarden worden uitgevoerd. Deze *precondities* kunnen o.a. worden gebruikt als toegangscontrole van een bepaalde stap (denk hierbij bijvoorbeeld aan het selecteren van de patiënten waarop de stap van toepassing is) of als een temporele eis (bijvoorbeeld de stap mag pas starten als een andere stap een bepaalde tijd geleden is beëindigd).

Voortijdig beëindigen van stappen

Soms is het niet mogelijk, wenselijk of noodzakelijk dat stappen volledig worden uitgevoerd. Het kan bijvoorbeeld nodig zijn een medische handeling af te breken vanwege het optreden van complicaties. Een ander voorbeeld is een medische handeling die voortijdig wordt beëindigd omdat het gewenste doel reeds bereikt is. Aan een stap kunnen dus condities verbonden zijn die aangeven wanneer de stap voortijdig beëindigd mag worden.

Postcondities

Postcondities zijn in deze context beweringen die gelden nadat een bepaalde stap met succes is uitgevoerd. Voorbeeld hiervan is dat na meting van de bloeddruk de bloeddruk bekend is.

Temporele kennis

Aan de stappen in een richtlijn kunnen temporele eisen verbonden zijn. Er kan voor een bepaalde stap bijvoorbeeld beschreven zijn wat de maximale duur is of wanneer de stap moet beginnen (in het laatste geval wordt de stap ‘vertraagd’ gestart: er wordt een bepaalde tijd gewacht alvorens de stap uitgevoerd wordt). Dikwijls wordt er in richtlijnen dan ook aan tijdstippen gerefereerd, zoals het begin van een bepaalde stap of het bekend worden van bepaalde informatie. Hierbij wordt er veelal onderscheid gemaakt tussen verschillende tijds-eenheden, zoals seconden, uren, dagen, enz. Vaak bestaat er onzekerheid omtrent temporele

kennis, d.w.z temporele kennis is niet altijd exact beschreven. Voor een bepaalde stap kan bijvoorbeeld worden geëist dat deze minimaal 1 dag en maximaal 3 dagen na een andere stap moet starten, i.p.v. dat de stap exact 2 dagen na de andere moet starten.

Representatie van data

In een richtlijn wordt vaak naar (patiënt)gegevens verwezen, zoals leeftijd, geslacht, bloeddruk, enzovoort. Het kan hier gaan om ‘ruwe’ data die rechtstreeks afkomstig is van een medische onderzoek of aan de patiënt wordt gevraagd. Het kan echter ook om gegevens gaan die afgeleid zijn van andere gegevens; het gegeven dat een patiënt hypertensief is, is bijvoorbeeld afgeleid van de hoogte van de bloeddruk. In het laatste geval spreekt men van *data-abstractie*. Met data wordt meestal ook een type en een eenheid geassocieerd. Leeftijd is bijvoorbeeld van het type ‘getal’ en wordt uitgedrukt in ‘jaren’. Het type en de eenheid van data is vaak niet expliciet vastgelegd in een richtlijn. Verder kan het voorkomen dat data op verschillende tijdstippen verschillende waarden kan aannemen; denk hierbij bijvoorbeeld aan de bloeddruk van een patiënt die meerdere malen gemeten wordt.

2.3 Representatie van richtlijnen: modelleertalen

Richtlijnen worden doorgaans als relatief ongestructureerde documenten gerepresenteerd, waarbij het gebruikte formaat per richtlijn kan verschillen. In deze documenten zijn vaak de meest uiteenlopende representaties te vinden, zoals tekst, tabellen, lijsten, stroomdiagrammen, graven, enzovoort. Vaak komt het hierbij voor dat richtlijnen hiaten en dubbelzinnigheden bevatten, omdat de auteur van de richtlijn details over het hoofd ziet of bepaalde kennis als algemeen bekend veronderstelt. Het representeren van richtlijnen in een gemeenschappelijk, formeel formaat kan ten opzichte van de informele representatie vier belangrijke voordelen bieden ([2, 9]):

- Een richtlijn gerepresenteerd in een formeel formaat maakt het beter toepasbaar voor beslissings-ondersteunende systemen. Dergelijke systemen assisteren de arts door bijvoorbeeld automatisch de volgende stap in de richtlijn te selecteren, het controleren van gegevens op consistentie en volledigheid en het geven van ‘reminders’. Tevens kan de complexiteit van een richtlijn gemaskeerd worden door slechts enkele, op dat moment relevante, taken van de richtlijn uit te lichten. Vaak is in zulke applicaties ook aanvullend didactisch materiaal, zoals plaatjes, simulaties, enz. opgenomen.
- Een gemeenschappelijke representatie voor (alle) richtlijnen zal de ontwikkeling van beslissings-ondersteunende systemen voor het implementeren van richtlijnen stimuleren, daar deze applicaties nu hergebruikt kunnen worden.
- Het omzetten van een ‘handgeschreven’ naar een formeel formaat maakt eventuele onvolledigheden en incorrectheden duidelijk en dwingt af dat er aandacht aan wordt besteed.
- Het vele werk dat gemoeid is met het ontwikkelen van een richtlijn maakt het zinvol om de verspreiding van richtlijnen over verschillende instellingen en ziekenhuizen te bevorderen. Dit is aantrekkelijk omdat het met de huidige mogelijkheden van de informatietechnologie mogelijk is om centraal richtlijnen op te slaan die vanuit verschillende instituten en ziekenhuizen geraadpleegd en eventueel aangevuld kunnen worden. Een gemeenschappelijk en ondubbelzinnig formaat kan hierbij bijdragen tot een uniforme interpretatie van de richtlijnen.

Er zijn inmiddels reeds diverse *representatietalen* of *modelleertalen* voorgesteld als gemeenschappelijk formaat voor richtlijnen. Een belangrijke eigenschap van zulke talen is dat deze, zij het direct of indirect, geschikt zijn voor gebruik door de computer. De *Arden syntax* bijvoorbeeld, is een Pascal-gelijkende taal waarmee zgn. *Medical Logic Modules* (MLM's) ontwikkeld kunnen worden. Dit zijn software-modules die reageren op patiëntgegevens in databases, met bijvoorbeeld waarschuwingen en opmerkingen. Een belangrijk nadeel van de Arden syntax is dat de uitdrukingskracht van procedurele kennis beperkt is, waardoor de taak ongeschikt is voor het modelleren van complexe richtlijnen ([5]). Een ander voorbeeld is GLIF, een taal die ontstaan is als een consensus uit vier andere modelleertalen. Hierin wordt een richtlijn beschouwd als een verzameling acties, uit te voeren met branching-logic. Het grote nadeel van GLIF is dat de taal niet formeel genoeg is voor gebruik door een computer: veel kennistypen worden als vrije tekst gerepresenteerd ([5, 10]).

De twee modelleertalen die in deze scriptie vergeleken worden, PROforma en Asbru, lijken op het eerste gezicht geschikt om complexe richtlijnen te modelleren in een formele formaat. Aan PROforma en Asbru ligt een klinisch model van stappen, beslissingen, enzovoort ten grondslag. Doordat deze talen hiervoor concepten gebruiken die ook door richtlijnen en experts (lees artsen) worden gebruikt, zal dit de kennisacquisitie en het onderhoud verbeteren en vergemakkelijken.

Om verwarring te voorkomen wordt in de rest van de scriptie met *richtlijn* de 'ad hoc' geschreven richtlijn en met *protocol* de gemodelleerde versie van de richtlijn bedoeld.

Hoofdstuk 3

Beschrijving van PROforma

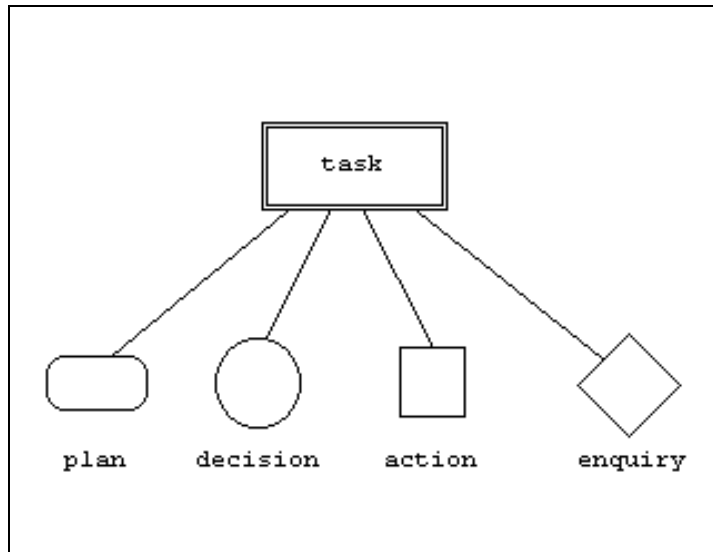
PROforma is ontwikkeld door het Imperial Cancer Research Fund te London en maakt deel uit van het PROMPT-project (PROtocols for Medical Procedures and Therapies). Tijdens het modelleren van de richtlijn in PROforma is gebruik gemaakt van Arezzo; dit is een tool waarmee in PROforma richtlijnen kunnen worden gemodelleerd en getest. De behandeling van PROforma betreft dan ook de versie die in Arezzo is opgenomen.

Het formaat waarin in PROforma richtlijnen worden gemodelleerd maakt het mogelijk om het verkregen protocol door een computer te laten uitvoeren. Tijdens het uitvoeren van een protocol wordt de gebruiker verzocht om bepaalde handelingen uit te voeren of gegevens te verzamelen. Ook kunnen op basis van de gegevens (automatisch) beslissingen worden genomen over de te nemen stappen. In dit hoofdstuk zullen de syntax en de semantiek van PROforma besproken worden, evenals de reeds beschikbare software.

3.1 PROforma ontologie

In de ontologie van PROforma vormt de *task* ofwel *taak* de basis ([1, 2]). Er worden vier klassen van taken onderscheiden (zie ook figuur 3.1):

- de *action* taak. Dit is een procedure die extern wordt uitgevoerd, zoals bijvoorbeeld het toedienen van een injectie of het voorschrijven van medicatie.
- de *enquiry* taak. Met deze taak worden patiëntgegevens die in de richtlijn worden gebruikt verkregen.
- de *decision* taak. De taak *decision* wordt gebruikt om (complexe) beslissingen te nemen. Deze beslissingen variëren van het stellen van een diagnose tot het kiezen van de volgende stap in de richtlijn.
- de *plan* taak. Deze bestaat uit een reeks deeltaken. In een plan wordt vastgelegd in welke volgorde deze deeltaken worden uitgevoerd. Ook kan hiermee een logische samenhang van taken worden uitgedrukt. In PROforma wordt een richtlijn als een plan gemodelleerd.



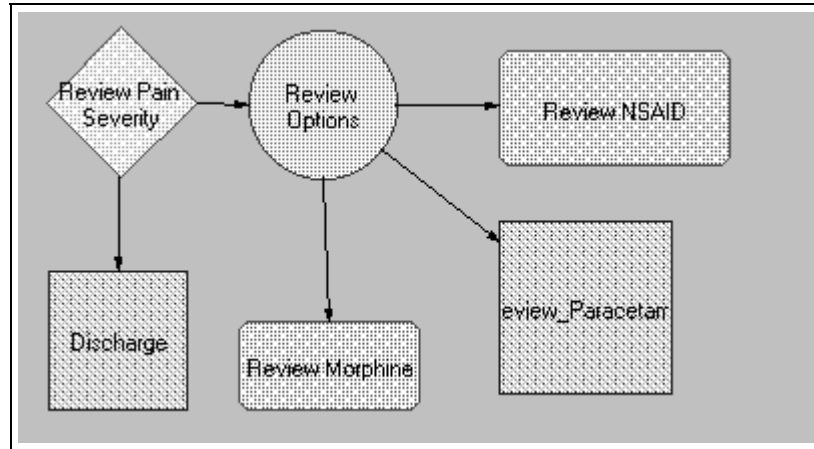
Figuur 3.1: De taak-klassen van PROforma .

De specificatie van een taak-klasse wordt uitgedrukt middels een verzameling attributen waarmee eigenschappen en het gedrag tijdens het uitvoeren van het protocol gedefinieerd kunnen worden. Een deel van deze attributen zijn *generieke* attributen, d.w.z. attributen die voor elk type taak gedefinieerd kunnen worden. Het resterende deel van de attributen zijn specifiek voor een taak-klasse. Elke taak kan bijvoorbeeld een *description* (beschrijving) of een *precondition* (preconditie) hebben, terwijl het attribuut *procedure* alleen bij de taak *action* voorkomt.

Naast de task kent PROforma nog een andere entiteit, nl. het *data-item*. Met deze data-items kunnen gegevens, die tijdens het uitvoeren van een protocol opgevraagd worden, worden gerepresenteerd. Net als een task heeft ook een data-item attributen; hierin kunnen bijvoorbeeld het data-type en mogelijke waarden voor het data-item gedefinieerd worden. Het modelleren in PROforma gebeurt in twee fasen: eerst worden in grote lijnen de taken en de volgorde tussen de taken gemodelleerd. Dit wordt meestal grafisch gedaan: de taken action, enquiry, decision en plan worden dan gerepresenteerd met resp. een vierkant, een ruit, een cirkel en een afgestompte rechthoek; de volgorde van de taken wordt met pijlen weergegeven. Vervolgens worden dan de attributen van de taken gedefinieerd, evenals de data-items. In de figuren 3.2 en 3.3 zijn resp. een grafische en tekstuele weergave te zien van een protocol.

3.2 Syntax en semantiek van PROforma

In paragraaf 3.2.1 worden per taak de attributen beschreven. Informatie over patiëntgegevens wordt opgeslagen in de attributen van *data-items*; deze zullen in paragraaf 3.2.2 aan de orde komen. In paragraaf 3.2.3 zal worden gekeken welke expressies er in attributen kunnen voorkomen.



Figuur 3.2: Grafische voorstelling van protocollen in *PROforma*. In dit voorbeeld is het plan *FollowUp* te zien, een deelplan van een protocol voor pijnbestrijding bij terminale kankerpatiënten; het bestaat uit twee actions, een enquiry, een decision en twee deelplannen. De pijlen geven de volgorde weer; de action *Discharge* bijvoorbeeld mag pas starten wanneer de enquiry *ReviewPainSeverity* beëindigd is.

```

plan :: Follow_Up ;
  caption :: 'Follow Up' ;
  description :: 'Follow up of initial drug treatment.' ;
  component :: e_Review_Severity ;
  component :: Review_Options ;
    schedule_constraint :: completed(e_Review_Severity) ;
  component :: a_Paracetamol ;
    schedule_constraint :: completed(Review_Options) ;
  component :: ReviewMorphine ;
    schedule_constraint :: completed(Review_Options) ;
  component :: ReviewNSAID ;
    schedule_constraint :: completed(Review_Options) ;
  component :: Discharge ;
    schedule_constraint :: completed(e_Review_Severity) ;
end plan .

```

Figuur 3.3: De attributen behorende bij plan *FollowUp* uit figuur 3.2.

3.2.1 Representatie van taken

Generieke attributen

De generieke attributen kunnen in alle taak-klassen voorkomen:

<i>Attribuut</i>	<i>Beschrijving</i>
Name	Dit is de unieke naam van de taak.
Caption	Dit attribuut vervangt <i>name</i> wanneer de naam op een leesbare manier moet worden weergegeven.
Description	Hiermee wordt een omschrijving gegeven van de taak.
Goal	Het doel van de richtlijn.
Precondition	De conditie die moet gelden voordat de taak start. Wanneer deze conditie niet geldt wordt de taak niet uitgevoerd.
Trigger condition	De gebruiker krijgt tijdens het uitvoeren van het protocol een lijst met triggercondities te zien. Wanneer de gebruiker één van deze condities bevestigt wordt de taak waarvoor de triggerconditie gedefinieerd is gestart, ongeacht de scheduling constraints.
Postcondition	Dit beschrijft een propositie die geldt na terminatie van de taak.
Optional	Hiermee kan worden weergegeven of een taak optioneel is, in tegenstelling tot verplicht. Dit geeft de gebruiker de optie om de taak over te slaan. Er kan gekozen worden uit yes en no .
Automatic	Deze optie geeft aan of de gebruiker toestemming moet geven voor uitvoering van de taak. Er kan gekozen worden uit yes en no .
Number of cycles	Aantal keren dat een taak moet worden uitgevoerd. Dit wordt uitgedrukt door een getal of door een expressie die tot een getal berekend kan worden.
Cycle until	Dit is een conditie. Het geeft aan dat de taak herhaald moet worden totdat hieraan voldaan is.
Cycle interval	Hiermee kan worden weergegeven hoeveel tijd er tussen twee opeenvolgende uitvoeringen van taken moet zitten.
Parameters	Deze zijn vergelijkbaar met formele parameters uit programmeertalen en zijn lokaal t.o.v. de taak. De actuele waarde van de parameter is vastgelegd in het attribuut <i>parameter values</i> in het <i>plan</i> waarvan de taak deel uitmaakt. Een parameter kan worden gebruikt in de andere attributen van de taak.

Attributen van een action

Bij de taak *action* wordt de opdracht gegeven tot een externe handeling. Vanuit PROforma wordt dit ondersteund door de arts een beschrijving te geven van de uit te voeren handeling. Naast de generieke attributen zijn de volgende attributen mogelijk:

<i>Attribuut</i>	<i>Beschrijving</i>
Procedure	Een beschrijving van de uit te voeren handeling.
Context	Aanvullende informatie voor het uitvoeren van de handeling.

Attributen van een enquiry

Tijdens de taak *enquiry* worden patiëntgegevens verzameld. Naast de generieke attributen is het volgende attribuut mogelijk:

<i>Attribuut</i>	<i>Beschrijving</i>
Sources	Hierin worden de gegevens opgesomd die tijdens de enquiry opgevraagd worden.

Attributen van een decision

Een *decision* neemt een beslissing uit een reeks kandidaten. De beslissing is hierbij gebaseerd op het aantal voor- en tegenargumenten voor een kandidaat. Naast de generieke attributen zijn de volgende attributen mogelijk:

<i>Attribuut</i>	<i>Beschrijving</i>
Candidates	Beschrijft de kandidaten waaruit gekozen kan worden.
Arguments	Voor elke kandidaat kunnen hier de argumenten voor of tegen worden ingevuld, alsmede de kracht (zie volgende attribuut <i>support mode</i>) van deze argumenten.
Support mode	Hiermee kan worden weergegeven op welke manier de argumenten gewaardeerd worden. Bij de optie <i>symbolic support</i> kan een argument t.o.v. een kandidaat voor, tegen, bevestigend of afwijzend zijn. Bij de optie <i>numeric support</i> kunnen aan de argumenten (positieve en negatieve) gewichten worden toegekend.
Recommendation rules	Voor elke kandidaat kan met deze conditie worden aangegeven wanneer tot de kandidaat besloten wordt. De rol van de argumenten hierin kan worden uitgedrukt met de functie <i>netsupport</i> , die voor elke kandidaat de waarden van de argumenten sommeert.
Sources	Dit zijn de gegevens die gebruikt worden tijdens de taak <i>decision</i> .
Multiple choice	Met deze optie kan worden aangegeven of er één of meer dan één kandidaten gekozen kunnen worden.

Attributen van een plan

Een *plan* wordt gebruikt om van een groep taken de volgorde tussen en de samenhang van deze taken te modelleren. Naast de generieke attributen zijn de volgende attributen mogelijk:

<i>Attribuut</i>	<i>Beschrijving</i>
Components	Dit zijn de deeltaken van de taak <i>plan</i> .
Scheduling constraints	Deze geven de volgorde aan waarin de taken moeten worden uitgevoerd.
Abort condition	Wanneer deze conditie geldt wordt de taak <i>plan</i> afgebroken.
Termination condition	Wanneer deze conditie geldt wordt de taak <i>plan</i> direct beëindigd.
Parameter Values	Dit is een assignment waarmee aan een parameter van een taak een waarde kan worden toegekend.

3.2.2 Representatie van data-items

Data-items kunnen worden gezien als globale variabelen; ze bestaan uit de volgende attributen:

<i>Attribuut</i>	<i>Beschrijving</i>
Name	Dit is de unieke naam van het data-item.
Caption	Dit attribuut vervangt <i>name</i> wanneer de naam op een leesbare manier moet worden weergegeven.
Description	Hiermee wordt een omschrijving gegeven van het data-item.
Type	Het datatype van het data-item.
Range	Een verzameling waarden die het data-item kan aannemen.
True value	Dit is de waarde die bij een boolse data-item als <i>True</i> geldt.
False value	Dit is de waarde die bij een boolse data-item als <i>False</i> geldt.
Default value	Deze waarde wordt de gebruiker aanbevolen tijdens uitvoering van het protocol. Dit kan ook een expressie zijn die een waarde oplevert.
Mandatory validation	Dit is een conditie waar de waarde van het data-item aan moet voldoen.
Warning condition	Een conditie m.b.t. de waarde van het data-item. Wanneer deze geldt wordt er een waarschuwing gegeven.
Derivation rule	Een afleidingsregel voor de waarde van het data-item. Wanneer deze leeg is wordt de waarde van het data-item aan de gebruiker gevraagd.
Unit	De eenheid van de waarde van het data-item.

3.2.3 Expressies in attributen

De taal *PROforma* beschikt over een reeks data-representaties en operatoren waarmee complexe expressies en condities kunnen worden opgebouwd. Deze expressies en condities worden gebruikt in de attributen van taken en data-items. De volgende expressies zijn mogelijk:

Temporele expressies voor het representeren van tijd

Met de datastructuren *date*, *time* en *datetime* kunnen constante tijdstippen worden gerepresenteerd. De functie *now* geeft het huidige tijdstip weer. Daarnaast is het mogelijk om met de functies *start* en *end* aan de begin- en eindtijdstippen van taken te refereren. Met de functie *dateof* tenslotte kan worden gerefereerd aan het tijdstip waarop een data-item een waarde heeft gekregen.

Data-items

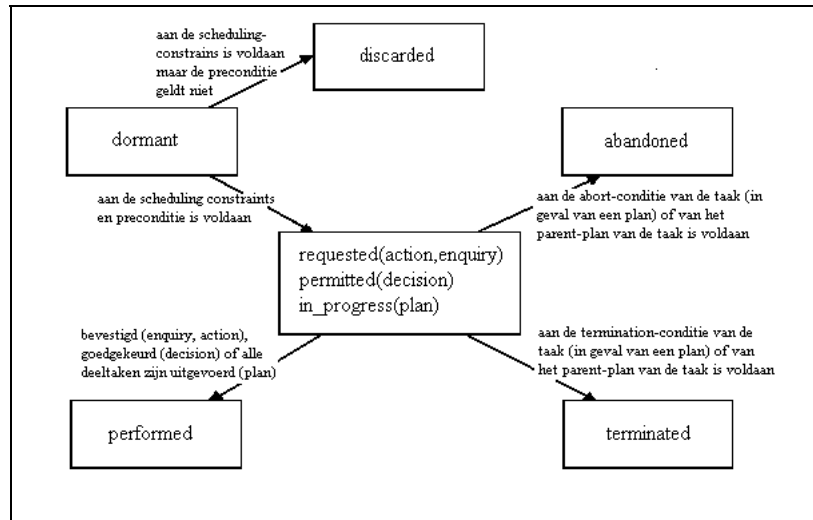
In *PROforma* kan een data-item tijdens het uitvoeren van het protocol meer dan één waarde verkrijgen, waarbij bij elke waarde ook het tijdstip wordt opgeslagen waarop de waarde bekend is geworden. Wanneer in een expressie een data-item voorkomt wordt hiermee normaal gesproken naar de laatste waarde verwezen; referentie naar eerdere waarden is mogelijk met de verzamelfuncties *first*, *last*, *any* en *every* of door het tijdstip van bekend worden te vergelijken met andere tijdstippen m.b.v. de functies *during*, *before*, *after*.

Numerieke expressies

Proforma kent numerieke operatoren zoals +, -, *, enz. en functies die een verzameling (met numerieke expressies) als argument hebben, nl. *count*, *average*, *sum*, *max* en *min*. De functie *datediff* berekent het verschil tussen twee tijdstippen. Tenslotte is er de functie *netsupport*, deze is reeds bij de beschrijving van de taak *decision* aan de orde geweest.

Boolse expressies

In *PROforma* kunnen expressies worden gebouwd die bij evaluatie (lees: bij het uitvoeren van de richtlijn) de waarde TRUE of FALSE opleveren. *PROforma* kent numerieke vergelijkingsoperatoren (=, >, >=, enz.), temporele vergelijkingsoperatoren



Figuur 3.4: De verschillende stadia van *PROforma*. Initieel is een taak in toestand *dormant*. Nadat er aan de scheduling-constraints is voldaan wordt de preconditione getest. Wanneer deze niet geldt wordt de taak afgewezen (*discarded*), wanneer deze wel geldt wordt de taak geactiveerd. De taak kan vervolgens voltooid zijn (*performed*), voortijdig afgebroken (*abandoned*) of voortijdig beëindigd (*terminated*).

(*isbefore, isafter, isduring*), vergelijkingoperatoren voor verzamelingen (*include, oneof*) en vergelijkingoperatoren voor tekst waarmee twee stukken tekst op gelijkheid kunnen worden getest. Met de functie *isknown* kan worden getest of een data-item een waarde heeft. Daarnaast kunnen met de logische operatoren *and, or* en *not* samengestelde condities worden opgebouwd.

If-then-else expressie

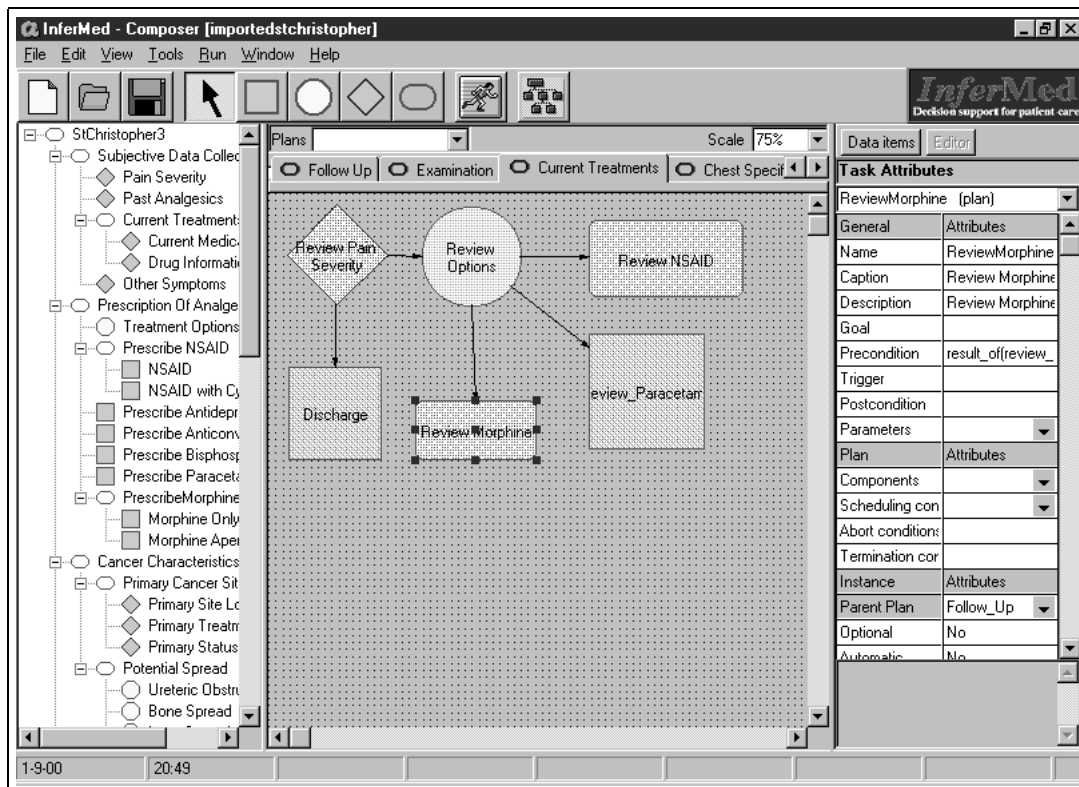
Met een *If-then-else*-expressie, vergelijkbaar met if-statements uit de meeste talen, kan uit twee alternatieve expressies worden gekozen waarbij de keuze afhankelijk is van een conditie.

De functie *result_of*

De functie *result_of* heeft als argument een *decision* en als resultaat de kandidaat waartoe in de *decision* is besloten.

3.3 Uitvoering van een protocol

Tijdens het uitvoeren van een protocol kunnen de taken diverse toestanden doorlopen. Hoewel deze toestanden niet expliciet een onderdeel zijn van de taal is het toch zinvol om de toestanden hier te behandelen omdat in de attributen van taken soms naar deze toestanden verwezen wordt. Initieel zijn taken *dormant*, of vrij vertaald 'in rust', wat betekent dat ze nog niet gestart zijn. Een taak waarvan de scheduling constraints en preconditione gelden wordt geactiveerd. De taken *action* en *enquiry* komen in dit geval in de toestand *requested*, de taak *decision* in de toestand *permitted*, en een *plan* in toestand *in_progress*. De taken *action*, *enquiry*, *decision* en *plan* zijn uitgevoerd (*performed*) wanneer resp. de procedure van de *action* door de gebruiker bevestigd is, de data-items van de *enquiry* bekend zijn, een kandidaat van de *decision* door de gebruiker goedgekeurd is en de deeltaken van een plan uitgevoerd

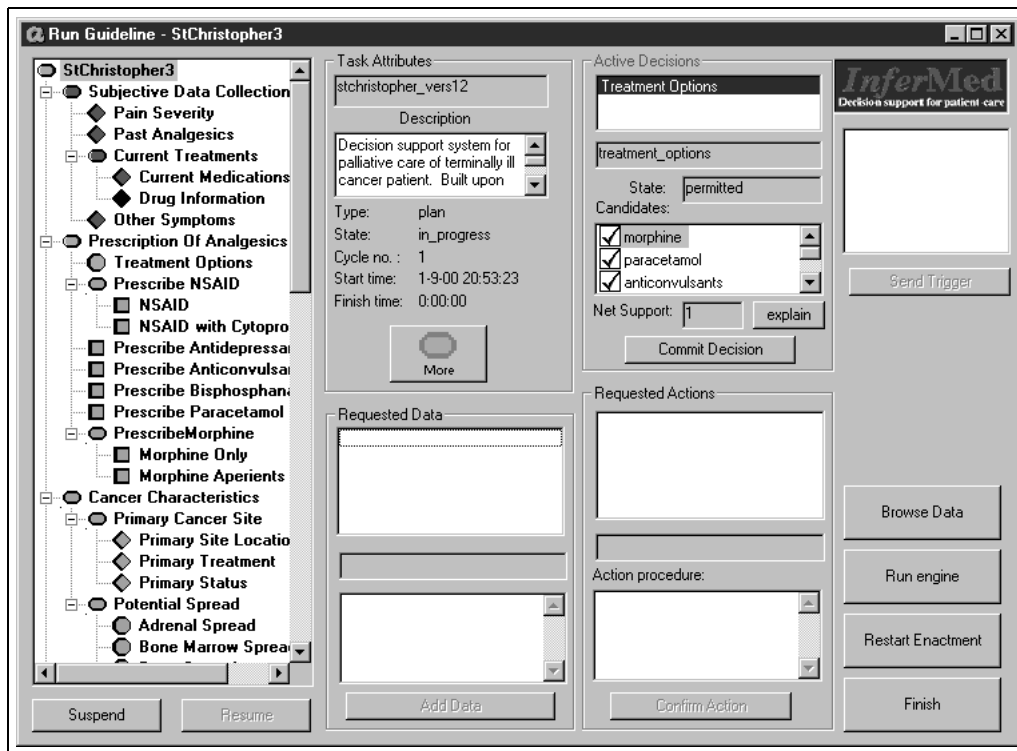


Figuur 3.5: Het ontwikkelen van een protocol met behulp van *Composer*, in dit voorbeeld het plan *FollowUp*. Links in de figuur is een hiërarchisch overzicht te zien van het protocol waar het plan deel van uitmaakt. In het midden zijn de onderdelen van het plan te zien. Aan de rechterkant is een window te zien waarin aan de attributen van plan *ReviewMorphine* een waarde kan worden toegekend.

zijn. Een geactiveerd plan kan daarnaast nog in twee andere toestanden eindigen: toestand *terminated* in geval de *termination-condition* waar wordt, en toestand *abandoned* wanneer de *abort-conditie* geldt. Ook de reeds geactiveerde deeltaken van het plan komen in deze gevallen in de respectievelijke toestanden *terminated* en *abandoned*. Een taak die op grond van de scheduling constraints geactiveerd zou moeten zijn maar niet aan de preconditione voldoet wordt afgewezen (*discarded*). In figuur 3.4 zijn de verschillende stadia te zien van een taak.

3.4 Softwaretools voor PROforma : Arezzo

Voor de taal PROforma is, zoals eerder vermeld, het softwarepakket *Arezzo* beschikbaar; dit pakket bestaat uit de *Arezzo Composer* en de *Arezzo Tester*. De Composer vergemakkelijkt het modelleren van richtlijnen. Een richtlijn kan in grote lijnen beschreven worden door een grafische representatie op verschillende niveaus. Hiermee worden de taken en de scheduling constraints gemodelleerd. Voor de taken in de graaf zijn vervolgens *templates* beschikbaar waar voor de desbetreffende attributen de waarden kunnen worden ingevuld. In figuur 3.5 is een voorbeeld te zien van het modelleerproces in Arezzo.



Figuur 3.6: Het uitvoeren van het protocol uit figuur 3.5 in Tester. In dit fragment wordt de decision Treatment Options uitgevoerd.

De in Composer ontwikkelde protocollen kunnen in de Tester getest worden. Hiermee kan worden gekeken hoe een protocol zich gedraagt wanneer het uitgevoerd wordt. Tijdens het uitvoeren van een protocol wordt begonnen met het plan op het hoogste niveau: deze krijgt de status *in_progress*; de taken in dit plan doorlopen vervolgens de verschillende stadia zoals die in de vorige paragraaf behandeld zijn. Eventuele benodigde gegevens en bevestigingen die nodig zijn voor het uitvoeren van de taken worden aan de gebruiker gevraagd. In figuur 3.6 is een voorbeeld te zien van de uitvoering van een protocol.

Hoofdstuk 4

Beschrijving van Asbru

De taal Asbru (in dit onderzoek is versie 7.2 gebruikt) maakt deel uit van het Asgaard-project en is o.a. ontwikkeld door de Universiteit van Wenen. De belangrijkste structuur in Asbru is het element *plan*, vergelijkbaar met een *stap* uit hoofdstuk 2 en een *taak* in Proforma. Een richtlijn wordt in Asbru gerepresenteerd als een verzameling van zulke plannen. Paragraaf 4.2 zal dieper ingaan op de semantiek van het element plan en andere elementen uit de taal. Om beter inzicht te krijgen in de taal zullen echter eerst in paragraaf 4.1 enkele belangrijke concepten van Asbru behandeld worden. In paragraaf 4.3 zal worden besproken hoe een richtlijn in Asbru gemodelleerd wordt.

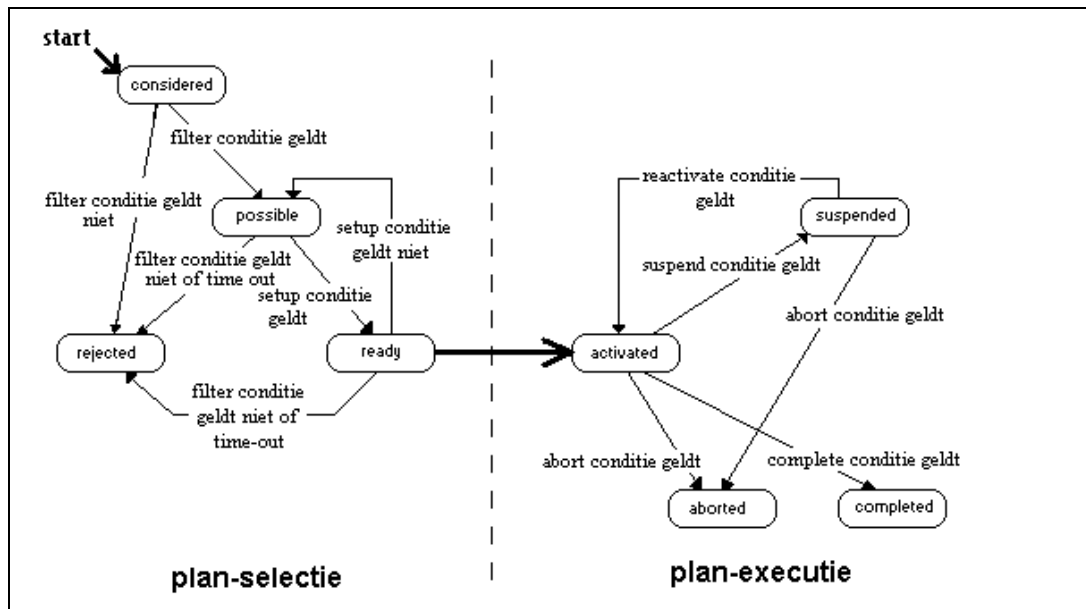
4.1 Asbru als onderdeel van het Asgaard-project

De taal Asbru maakt deel uit van het Asgaard-project ([4, 5, 6, 7]). In dit project wordt getracht de handelingen van een arts te ondersteunen door het beschrijven van een verzameling taken ([4]), zoals het bepalen van de volgende stap in een richtlijn, het herkennen van de intentie van de arts en het beoordelen van de handelingen van de arts in het licht van de richtlijn en beschikbare patiëntgegevens. Deze taken moeten niet verward worden met een *taak* in Proforma of een richtlijn-stap. Het zijn geen elementen van de taal in de zin dat er inhoudelijke richtlijnkennis mee wordt gerepresenteerd, maar meta-taken die het geheel van ontwikkeling, uitvoering en evaluatie van een gemodelleerde richtlijn beschrijven. De kennis die nodig is voor deze taken worden *knowledge roles* genoemd, en verwijst naar plan-attributen en patiëntgegevens.

Een belangrijke eigenschap van Asbru is dat de auteur van een plan de intentie kan weergeven; met behulp van intelligente systemen kunnen dan de handelingen en beslissingen van een arts vergeleken worden met de stappen die het protocol aandraagt. Zulke intenties zijn vooral bruikbaar wanneer de klinische toestand van een patiënt bewaakt moet worden.

4.2 De semantiek van Asbru-elementen

Deze paragraaf bespreekt de semantiek van Asbru. Het belangrijkste element in Asbru is het plan, en zal in 4.2.1 aan de orde komen. Daarop volgend zullen enkele essentiële elementen die in plannen voorkomen behandeld worden. Vanwege de complexiteit van de syntax (zie 4.3) zijn de voorbeelden veelal in vrije tekst gegeven.



Figuur 4.1: De plan-toestanden in Asbru. Een plan die voor activatie in aanmerking komt heet *considered*. Wanneer de *filter-conditie* geldt komt het plan in toestand *possible*. Indien de *setup-conditie* gedurende een bepaalde tijd geldt, komt het plan in toestand *ready*. Merk op dat de filter-conditie en de setup-conditie tijdens de plan-selectie weer onwaar kunnen worden. Een plan is *rejected* wanneer de filterconditie in het begin niet geldt of tijdens toestand *possible* of *ready* weer onwaar wordt, of wanneer de tijd, gedurende welke de setup-conditie moet gelden, is verstreken. Een plan wordt vervolgens, of automatisch, of na bevestiging van de gebruiker, geactiveerd (toestand *activated*). De overgang naar andere toestanden is dan afhankelijk van de *abort*-, *complete*- en *suspendcondities*.

4.2.1 Plan

Tijdens het uitvoeren van een protocol kunnen de daartoe behorende plannen verschillende *toestanden* doorlopen. Daarbij kan er een onderscheid gemaakt worden tussen toestanden tijdens de selectie-fase, waarin gekeken wordt of een plan aan de precondities voldoet om uitgevoerd te worden, en toestanden tijdens de executie-fase. In figuur 4.1 zijn deze toestanden grafisch weergegeven.

Een plan beschrijft een verzameling stappen waarmee een bepaald doel, vastgelegd als *intentions* of *effects*, bereikt dient te worden. Deze stappen en de volgorde tussen deze stappen zijn vastgelegd in de *plan-body*. De overgangen van de verschillende toestanden worden gestuurd door een groep *conditions*, en eventuele voorkeuren bij het selecteren van het plan kunnen worden gerepresenteerd door *preferences*. De bovengenoemde onderdelen van een plan komen voor als de reeds eerder genoemde knowledge-roles, en zullen hieronder in detail besproken worden.

Preferences: Hierin is informatie gerepresenteerd die gebruikt wordt bij het selecteren van plannen. Voorbeeld hiervan zijn de ‘kosten’ van het uitvoeren van het plan; dit kan een letterlijk bedrag zijn, maar bijvoorbeeld ook het ongemak voor de patiënt. Een ander voorbeeld is de algemene strategie van het plan (agressief of normaal).

Intentions: Hiermee wordt het doel van het plan gespecificeerd. Intentions zijn *temporal patterns* (deze zullen verderop behandeld worden) die gehandhaafd, bereikt of vermeden

(in Asbru-syntax **maintain**, **achieve** of **avoid**) moeten worden. In deze temporal patterns wordt verwezen naar patiëntgegevens en/of acties van de arts. Asbru onderscheidt vier typen intenties:

- **intermediate state**
De toestand die tijdens het uitvoeren van een plan moet gelden.
 - **intermediate action**
De stappen of handelingen die gedurende de uitvoering van een plan gedaan moeten worden.
 - **overall state pattern**
De toestand die na beëindiging van een plan moet gelden.
 - **overall action pattern**
De stappen of handelingen die na beëindiging van een plan gedaan moeten zijn.
- Voorbeeld van een intention van het type overall action pattern is: **gedurende het plan is dagelijks de temperatuur gemeten.**

Conditions: Dit zijn, net als intentions, temporal patterns. Deze condities sturen de toestandsveranderingen van het plan. Asbru kent de volgende condities:

- **filter-precondition**
Wanneer het plan overwogen wordt (plan-toestand *considered*) moet deze conditie gelden om over te gaan naar plan-toestand *possible*.
- **setup-precondition**
Deze conditie beschrijft een toestand die bereikt moet worden om een plan te starten. Wanneer deze toestand geldt gaat de plan-toestand over van *possible* in *ready*. Een voorbeeld hiervan is dat **voor aanvang van een plan de patiënt 12 uur gevast heeft.**
- **activate-mode**
Hiermee kan worden aangegeven of een plan met plan-toestand *ready* automatisch of pas na toestemming van de gebruiker gestart wordt (plan-toestand *activated*).
- **suspend-condition**
Wanneer deze conditie geldt wordt de executie van het plan tijdelijk opgeschort (plan-toestand *suspended*). Het plan wordt weer actief wanneer de *reactivate-condition* geldt. Voorbeeld: **temperatuur = hoog**
- **reactivate-condition**
Wanneer deze conditie geldt wordt een tijdelijk opgeschort plan weer actief. Voorbeeld: **temperatuur = normaal**
- **complete-condition**
Hierin wordt vastgelegd wanneer een plan voltooid is.
- **abort-condition**
Wanneer deze conditie geldt wordt de executie van het plan afgebroken.

Effects: Hierin wordt het effect uitgedrukt dat het plan op patiëntgegevens (in dit geval parameters) heeft. Een voorbeeld hiervan is een plan dat **het glucose-niveau verlaagt met een waarschijnlijkheid van 80 procent en met een zekere context.**

Plan-body: Een plan-body specificeert de stappen van het plan. Dit kan een enkelvoudige stap zijn, een stap die een bepaald aantal keren moet worden herhaald of een complexe stap bestaande uit deelstappen. De stappen kunnen elementaire acties zijn zoals het

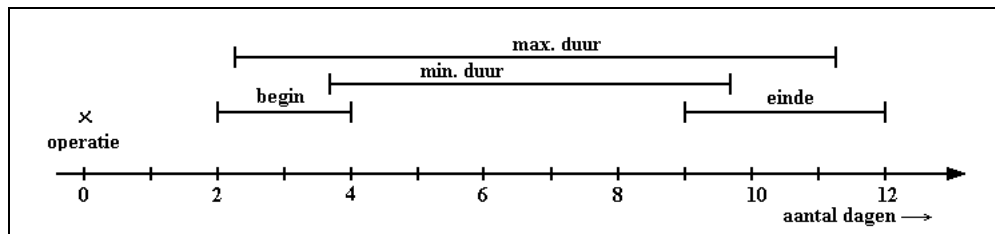
opvragen van patiëntgegevens of een if-then-else-statement, maar ook verwijzingen naar andere plannen. In het geval van complexe stappen zijn er vier mogelijke volgorden: *sequentieel*, *parallel*, *any-order* (er bestaat geen volgorde voor het uitvoeren van de stappen; wel wordt als extra eis gesteld dat telkens slechts één stap actief mag zijn) en *unordered* (er bestaat geen volgorde voor het uitvoeren van de stappen).

Bij een plan met een als complexe stap beschreven plan-body kan met een *continuation specification* worden aangegeven op welke deelplannen gewacht moet worden om het hoofdplan te beëindigen. Op deze manier wordt er in een complexe stap een onderscheid gemaakt tussen optionele en verplichte plannen: plannen zijn verplicht wanneer ze voorkomen in de continuation specification, anders zijn ze optioneel. Daarnaast is het met de optie *wait-for-optional-children* mogelijk om weer te geven of, wanneer de complete-condition waar wordt en alle verplichte plannen voltooid zijn, er op optionele plannen gewacht moet worden.

Het is mogelijk met een plan een time-annotation (zie 4.2.2) te associëren, die aangeeft wanneer en hoe lang het plan moet worden uitgevoerd. Daarnaast kunnen aan plannen argumenten meegegeven worden.

4.2.2 Time-annotations

Met behulp van zgn. *time annotations* (tijd-aantekeningen) is het mogelijk onzekerheid over een tijd-interval uit te drukken. Deze time-annotations worden gebruikt om de duur van een plan te specificeren of om bijvoorbeeld voor een bepaalde bewering te specificeren wanneer en hoe lang deze moet gelden. Een time-annotation beschrijft een tijd-interval, relatief aan een referentiepunt. Voor een tijd-interval hoeft dan niet het exacte begin- en eindtijdstip te worden vastgelegd. Er kan worden aangegeven dat de begin- en eindtijdstippen zelf binnen een tijd-interval liggen; daarnaast kan voor een tijd-interval de minimale en maximale duur worden gerepresenteerd. Figuur 4.2 geeft een schematische illustratie van een time-annotation in Asbru.



Figuur 4.2: Voorbeeld van een time-annotation. Het interval begint tussen 2 en 4 dagen en eindigt tussen 9 en 12 dagen na de operatie. Het interval duurt minimaal 6 en maximaal 9 dagen.

4.2.3 Temporal patterns (condities)

Intenties en condities worden uitgedrukt m.b.v. *temporal patterns*. Een temporal pattern is een bewering, gekoppeld aan een time-annotation die aangeeft wanneer en hoe lang de bewering geldt. Voorbeeld van een temporal pattern is de bewering **de temperatuur is niet hoger dan 39 graden** met de time-annotation **gedurende 3 dagen**. De beweringen in temporal patterns kunnen o.a. de volgende vorm hebben:

- Een *parameter-proposition*. Dit is een bewering over de waarde van een parameter (zie 4.2.4), in relatie met een context (zie 4.2.5). Met relationele operatoren zoals *less-than* en *equal* kan de waarde vergeleken worden met een expressie. Met een *value-range* kan worden getest of de waarde binnen bepaalde grenzen valt. Tenslotte kan worden getest of de parameter reeds een waarde heeft.
- Een *plan-state-constraint*. Deze doet een uitspraak over de toestand (zie figuur 4.1 voor de verschillende plan-toestanden) van een plan.
- Een temporele combinatie van temporal patterns, met temporele operatoren zoals *before* en *during*. Een voorbeeld hiervan is de bewering dat een parameter A eerder bekend is dan een parameter B.
- Een boolse combinatie van temporal patterns, met logische operatoren *and*, *or*, *xor* en *not*.
- Een *count-constraint*, waarmee uitgedrukt kan worden dat een temporal pattern een bepaald aantal keren moet gelden. Een voorbeeld hiervan is een parameter die, gedurende een bepaalde tijd, (minimaal) 5 keer gelijk moet zijn aan een bepaalde waarde.
- Een *simple-condition*. Dit zijn condities waarin operatoren uit temporal-patterns voorkomen, met het verschil dat een context en een time-annotation niet verplicht is, en dat het hier in het algemeen expressies betreft en niet alleen parameters. In Asbru omvatten expressies o.a. variabelen, parameters (zie 4.2.4 voor het verschil tussen variabelen en parameters), constanten, rekenkundige expressies en elementen uit lijsten en verzamelingen (zie 4.2.4).

4.2.4 Parameters en variabelen

Voor het representeren van data maakt Asbru onderscheid tussen *parameters* en *variabelen*. Beide kunnen van waarde veranderen, met het verschil dat bij een parameter naar de verschillende waarden verwezen kan worden. Dit is mogelijk omdat voor elke keer dat een parameter een waarde krijgt wordt bijgehouden wanneer deze waarde verkregen is.

Parameters en variabelen hebben een type. Dit kan een ingebouwd type zijn zoals een *string*, een *integer* of een *constante*. Daarnaast is het mogelijk om zelf een type te definiëren met een *qualitative-scale-def*, door de mogelijke waarden te noemen.

Voor een parameter kan o.a. worden aangegeven hoe vaak een nieuwe waarde gelezen of opgevraagd wordt en binnen welke grenzen de waarde moet vallen (alleen relevant bij numerieke waarden). Er zijn verschillende manieren waarop een parameter een waarde kan verkrijgen:

- Via een direct invoerkanaal.
- Berekend uit andere parameters en constanten, m.b.v. rekenkundige bewerkingen zoals optelling, aftrekking, vermenigvuldiging, worteltrekking, enz.
- Uit een reeks waarden van andere numerieke parameters, gemeten in een bepaald interval, kan analytische informatie verkregen worden zoals het gemiddelde, de helling en de standaard-afwijking.
- Een parameter verkrijgt zijn waarde middels data-abstractie, gebaseerd op een numerieke parameter. Het betreft hier *kwantitatieve* parameters, d.w.z. parameters van het type *qualitative-scale-def* waarbij met elke mogelijke waarde een interval is geassocieerd. Mogelijke waarden voor een parameter *hypertensie*, gebaseerd op een

parameter `systolische_bloeddruk`, zijn bijvoorbeeld `nee` en `ja` met resp. intervallen 0-140 en 141-oneindig.

Een variabele kan lokaal of globaal t.o.v. van een plan zijn, dit in tegenstelling tot parameters die altijd globaal zijn. Naast enkelvoudige waarden kan een variabele ook naar lijsten of verzamelingen verwijzen. Tijdens de definitie van een variabele is het mogelijk aan de variabele een beginwaarde toe te wijzen. Daarnaast kan een variabele via een *variable-assignment* in een plan-body een waarde verkrijgen. Voor een (numerieke) variabele zijn er opties waarmee de precisie en de afronding van een waarde weergegeven kunnen worden.

4.2.5 Context

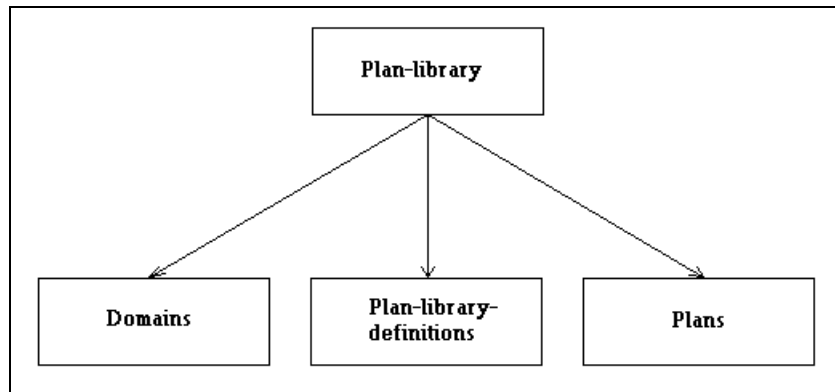
De data-abstractie en de evaluatie van temporal-patterns kan beïnvloed worden door een *context*. Het betreft hier een verzameling expressies gebaseerd op (boolse of kwalitatieve) variabelen en parameters. Een data-abstractie wordt dan bijvoorbeeld alleen toegepast wanneer de context geldig is.

4.3 Organisatie en syntax van Asbru

De gemodelleerde protocolkennis wordt in Asbru georganiseerd als een *plan-library*. Deze bestaat uit drie componenten (zie figuur 4.3). In *plans* worden de plannen gedefinieerd. De component *plan-library-definitions* bevat definities van functies, variabelen, constanten, time-annotations en types die in geheel de plan-library geldig zijn, onafhankelijk van een fysieke omgeving. De component *domain-definitions* beschrijft de onderdelen waarvan de specificatie afhankelijk is van de fysieke omgeving waarin de plannen worden uitgevoerd. Hierdoor is het mogelijk om een plan in verschillende omgevingen uit te voeren. Daarnaast kan de component domain-definitions ook voor verschillende plannen gebruikt worden. Naast onderdelen die domein-afhankelijk zijn, te weten context-definities, parameter-definities en iterator-definities (de laatste worden gebruikt bij lijsten), kan de component domain-definitions ook het soort onderdelen bevatten die in plan-library-definitions kunnen voorkomen.

In figuur 4.4 is een voorbeeld te zien van een plan-library, geschreven in de XML-syntax. In de XML-syntax wordt een plan-library gerepresenteerd als een boom-structuur van elementen. De specificatie van een element start met de naam van het element (tussen haakjes) en eindigt met een schuine streep gevolgd door de naam van het element (eveneens tussen haakjes). Hierbinnen kunnen *children* van het element opgenomen zijn. Daarnaast kan een element *attributes* bevatten; deze worden in het begin van de specificatie direct na de naam van het element geschreven.

Voor Asbru is vooralsnog geen software beschikbaar voor het ontwikkelen of testen van protocollen. Er is inmiddels wel het programma *AsbruView* beschikbaar waarmee visueel een indruk kan worden verkregen van de volgorde en temporele kennis van de plannen van een protocol.



Figuur 4.3: Organisatie van de plan-library.

<pre> <plan-library> <library-info title="Voorbeeld plan-library"> ... </library-info> <domain-defs> <domain> <parameter-groups> <parameter-def name="kwalitatieve-temperatuur" type="qual-temp"> <qualitative-parameter-def> <limits> <context> <any/> </context> <limit-entry value="0"/> <limit-entry value="36.8"/> <limit-entry value="37.9"/> <positive-infinite> </limits> <parameter-ref name="temperatuur"/> </qualitative-parameter-def> </parameter-def> ... </parameter-groups> </one-of-type-def> <qualitative-scale-def name="qual-temp"> <qualitative-entry entry="laag"/> <qualitative-entry entry="normaal"/> <qualitative-entry entry="hoog"/> </qualitative-scale-def/> ... </one-of-type-def> ... </domain> ... </domain-defs> <library-defs> ... </library-defs> <plans> ... </plans> </plan-library> </pre>	<pre> </parameter-def> ... </parameter-groups> </one-of-type-def> <qualitative-scale-def name="qual-temp"> <qualitative-entry entry="laag"/> <qualitative-entry entry="normaal"/> <qualitative-entry entry="hoog"/> </qualitative-scale-def/> ... </one-of-type-def> ... </domain> ... </domain-defs> <library-defs> ... </library-defs> <plans> ... </plans> </plan-library> </pre>
---	--

Figuur 4.4: Fragment van een (denkbeeldige) plan-library. Naast element `library-info` waarin aanvullende informatie en opmerkingen over de plan-library kan worden opgeslagen, bestaat de plan-library uit de drie elementen `domain-defs`, `library-defs` en `plans`.

Hoofdstuk 5

De cholesterolrichtlijn

Voor het vergelijken van Proforma en Asbru werd de cholesterolrichtlijn, zoals deze gebruikt wordt door specialisten aan het Universitair Medisch Centrum Utrecht (UMCU), gemodelleerd in beide talen. Deze gemodelleerde versies zijn gebaseerd op de consensus van het Centraal Begeleidingsorgaan voor de Intercollegiale Toetsing (CBO) van april 1998 ([12]), alsmede op informatie van medewerkers van het UMCU.

In paragraaf 5.1 zal de richtlijn globaal worden beproven. Paragraaf 5.2 behandelt de kennisbronnen die gebruikt zijn bij het acquireren en modelleren, en zal ingaan op de samenwerking met het UMCU. In paragraaf 5.3 tenslotte wordt een analyse gegeven van de richtlijn. Deze analyse is het uitgangspunt tijdens het modelleren en is deels gebaseerd op de consensus en deels op informatie die aangedragen werd door medewerkers van het UMCU.

5.1 De cholesterolrichtlijn: een introductie

Het doel van de cholesterolrichtlijn is het opsporen, diagnostiseren en behandelen van patiënten met een vetstofwisselingsstoornis. Een verhoogde cholesterolconcentratie, vooral een verhoogde *low-density*-lipoproteïnecholesterol (LDL)-concentratie, een verhoogde triglycerideconcentratie en een verlaagde *high-density*-lipoproteïnecholesterol (HDLc)-concentratie zijn belangrijke risicofactoren voor het optreden van coronaire hartziekten (CHZ). Daarnaast spelen ook andere factoren een belangrijke rol, zoals genetische aanleg, diabetes mellitus, hypertensie en roken.

De kern van de richtlijn is het zoeken op indicaties voor cholesterolverlagende geneesmiddelen (zgn. *statines*), en dit betekent in de praktijk dat er beslist moet worden of een patiënt moet worden behandeld met statines. Twee groepen patiënten komen eventueel in aanmerking voor behandeling met statines:

- patiënten met bekend vaatlijden (d.i. CHZ en perifeer of cerebraal vaatlijden), ook wel aangeduid met HVZ (hart- en vaatziekte). Voor deze patiënten worden in principe cholesterolverlagende geneesmiddelen voorgeschreven, mits er aan een aantal voorwaarden is voldaan met betrekking tot leeftijd, levensverwachting, enz. Screening en behandeling van deze groep wordt *secundaire preventie* genoemd.
- patiënten zonder bekend vaatlijden maar met een verhoogde kans op CHZ. De beslissing of dergelijke patiënten behandeld moeten worden is afhankelijk van de kans op ontwikkeling van CHZ in de eerstvolgende 10 jaar. Dit *10-jaarsrisico* kan worden

hoge bloed- druk	MANNEN																				
	Diabetes Nee										70 jaar	Diabetes Ja									
	Roken Nee					Roken Ja						Roken Nee					Roken Ja				
	ja	nee	ja	nee	ja	nee	ja	nee	ja	nee		ja	nee	ja	nee	ja	nee	ja	nee	ja	nee
20	25	29	32	35	28	33	38	42	45	25	30	34	38	41	34	39	44	48	51		
16	20	24	27	31	24	29	33	37	40	21	25	30	33	36	29	34	39	43	46		
14	18	21	24	27	21	25	30	33	36	18	22	26	30	33	26	31	35	39	42		
11	14	17	20	23	17	21	25	28	31	14	18	22	25	28	21	26	30	34	37		
8	11	14	17	19	14	17	21	24	27	12	15	18	21	24	18	22	26	30	33		
6	8	11	13	15	10	13	16	19	22	9	11	14	17	19	14	18	21	24	27		
4	5	7	9	11	7	9	12	14	16	6	8	10	12	14	10	13	16	19	21		
3	4	5	7	8	5	7	9	11	13	4	6	7	9	11	7	10	12	15	17		
	4	5	6	7	8	4	5	6	7	8	4	5	6	7	8	4	5	6	7	8	
Ratio totaal-cholesterol/HDL-cholesterol																					

Figuur 5.1: Onderdeel van de risicotabel (mannen).

afgelezen uit een tabel, en is gebaseerd op risicofactoren hypertensie, roken, diabetes mellitus, CHZ in familie, cholesterolconcentratie en leeftijd. Screening en behandeling van deze groep wordt *primaire preventie* genoemd.

Tijdens het opsporen wordt bepaald of een patient eventueel in aanmerking komt voor statines. Hierbij worden risicofactoren opgevraagd, zoals diabetes mellitus, roken, enz., en factoren die een patient uitsluiten van behandeling, zoals een levensverwachting kleiner dan vijf jaar. Bij het diagnostiseren worden bij de patienten die eventueel in aanmerking komen voor behandeling een lichamelijk onderzoek uitgevoerd. Het lichamelijk onderzoek bestaat uit het zoeken naar secundaire oorzaken voor hyperlipidemie en aanwijzingen op familiale hyperlipidemie, en het meten van de diverse cholesterolconcentraties. Vervolgens zullen eventuele secundaire oorzaken voor hyperlipidemie behandeld worden. Het diagnostiseren eindigt met de bepaling van het 10-jaarsrisico en de beslissing of de patient behandeld wordt. Deze beslissing is onder meer afhankelijk van het 10-jaarsrisico en verschillende drempelwaarden m.b.t. dit risico. Tijdens het behandelen worden statines voorgeschreven; het resultaat hiervan wordt na drie maanden geëvalueerd, waarbij de dosis eventueel verdubbeld wordt. Vervolgens wordt de patient jaarlijks gecontroleerd.

5.2 Kennisbronnen

5.2.1 CBO-consensus

Bij het modelleren van de richtlijn vormde de CBO-consensus “Behandeling en preventie van coronaire hartziekten door verlaging van de plasmacholesterolconcentratie” uit april 1998 ([12]) de basis. De kern van deze consensus, en het voor ons onderzoek belangrijkste onderdeel, bestaat uit een (samenvattende) richtlijn voor de behandeling en preventie van coronaire hartziekten, met daarbij een risicotabel voor personen zonder hart- en vaatziekten. In figuur

5.1 is een onderdeel van de risicotabel te zien. Deze richtlijn wordt in het resterende deel van de consensus toegelicht. Verschillende begrippen en definities uit de richtlijn (wanneer spreekt men bijvoorbeeld van *hyperlipidemie*?) worden hier verklaard. Daarnaast wordt hierin de context betreffende de richtlijn beschreven.

5.2.2 Samenwerking met het UMCU

Het modelleren van de cholesterolrichtlijn sloot aan bij *Second Manifestations of ARterial disease* (SMART) -onderzoek op het UMCU. Het hoofddoel van dit onderzoek is het vaststellen van prevalenties van ‘tweede’ vaatziekten en risicofactoren bij patiënten met een ‘eerste’ vaatziekte of risicofactor, d.w.z. reeds bekende vaatziekte of risicofactoren ([13]). Hierdoor kunnen hoog risicogroepen binnen de groep vaatpatiënten (of patiënten met een risicofactor) worden geïdentificeerd. Het onderzoek richt zich op patiënten die voor het eerst het UMCU bezoeken met een vaatziekte of een risicofactor. De voor het onderzoek benodigde metingen en onderzoeken worden verricht en de patiënten worden eventueel behandeld, waarin de toepassing van de cholesterolrichtlijn een rol speelt.

Binnen dit onderzoek was dan ook expertise aanwezig betreffende de cholesterolrichtlijn. In het kader van ons onderzoek werd ook deels op het UMCU gewerkt, wat de acquisitie van de expertise bevorderde. De medewerkers van het UMCU voelden zich betrokken bij ons project en waren bereid om hun medewerking te verlenen. De samenwerking en het contact met de medewerkers van het UMCU bestond uit het bijwonen van de wekelijkse SMART-vergaderingen waarin de algehele voortgang van het project werd besproken. Daarnaast werd er door het bijwonen van de wekelijkse ‘vaat-team’-besprekingen een indruk gekregen hoe behandeladviezen tot stand komen.

Met een specialist (dr. F. van Visseren) en een wetenschappelijk medewerkster (drs. W. Klein) was de samenwerking intensief. Er werden gesprekken gevoerd over hoe de richtlijn te interpreteren en op welke punten de richtlijn aangevuld diende te worden, waarbij niet alle gesprekken een formeel karakter hadden. Om een goed beeld te krijgen van het uitvoeren van de richtlijn in de praktijk, werd door hen beschreven hoe specialisten de CBO-richtlijn reeds gebruiken. Daarnaast werd samen met hen tijdens de ontwikkeling van de protocollen de voorlopige versies geëvalueerd m.b.t. volledigheid en correctheid. Het in de taal Proforma ontwikkelde protocol werd bovendien beoordeeld op gebruiksvriendelijkheid, omdat het uitvoeren van dit protocol met de in Arezzo beschikbare tools getest kon worden. De resultaten hiervan zullen in hoofdstuk 8 worden besproken.

5.3 Analyse van de richtlijn

Aan de hand van de CBO-richtlijn en de informatie van de experts werd er getracht een duidelijke specificatie van de richtlijn op te stellen. Hierbij week de informatie van de experts deels af van die in de CBO-richtlijn. De belangrijkste redenen hiervoor zijn:

- De CBO-richtlijn is op bepaalde punten niet exact genoeg beschreven. Dit laat de specialisten van het UMCU, die aan de hand van de richtlijn handelen, de ruimte om naar eigen inzicht de richtlijn te interpreteren. Om een volledige specificatie te verkrijgen werd de richtlijn aangevuld volgens de interpretatie van de specialisten.

- De CBO-richtlijn is niet specifiek voor specialisten opgesteld: ook bijvoorbeeld huisartsen maken er gebruik van. Dit komt tot uiting in enkele passages in de CBO-richtlijn waarin bij bepaalde bevindingen de patiënt naar een specialist moet worden verwezen. Omdat de CBO-richtlijn op het UMCU in de regel reeds door specialisten wordt uitgevoerd, konden deze punten niet ongewijzigd worden overgenomen. Besloten is dan ook om op deze punten de richtlijn volgens de werkwijze van de specialisten te modelleren.

In bijlage A zal een analyse worden gegeven van de richtlijn, waarop de protocollen in *PROforma* en Asbru gebaseerd zijn. Deze specificatie is beschreven in volgorde van de stappen en beslissingen die genomen moeten worden. Voor de punten waarop de specificatie afwijkt van de CBO-richtlijn zal een verklaring worden gegeven voor het verschil.

Hoofdstuk 6

Modelleren van de cholesterolrichtlijn in PROforma

Bij het modelleren van de richtlijn in PROforma is gebruik gemaakt van *Arezzo Composer*. Dit is een PROforma-editor waarin de taken van een richtlijn grafisch kunnen worden weergegeven met behulp van templates. Een bijbehorende tool *Arezzo Tester* is gebruikt om het met Composer verkregen protocol te testen. In paragraaf 6.1 zal bekeken worden hoe de kennistypen van een richtlijn, beschreven in hoofdstuk 2, in PROforma uitgedrukt kunnen worden. Deze kennistypen hoeven niet per se in de cholesterolrichtlijn voor te komen. Daarna zal in paragraaf 6.2 het protocol beschreven worden, met daarbij een motivatie van de keuzes die hier gemaakt zijn. Het modelleren van de richtlijn in PROforma bracht tenslotte enkele aspecten van PROforma aan het licht die in paragraaf 6.3 aan de orde zullen komen.

6.1 Representatie van de kennistypen in medische richtlijnen in PROforma

Doel

Voor elke taak is het in PROforma middels het attribuut *goal* mogelijk om het doel te modelleren. Het attribuut *goal* dient hier een boolse expressie te zijn vanwege de rol die het speelt bij de activering van de taak: wanneer de *goal* reeds geldt wordt de taak overgeslagen, en krijgt de status *performed*.

Atomaire stap

PROforma kent in feite drie basisconstructies waarmee eenvoudige acties van een arts gemodelleerd kunnen worden, nl. de *action* waarmee ‘externe’ handelingen worden gemodelleerd, de *enquiry* waarmee gegevens wordt verzameld en de *decision* waarmee beslissingen worden genomen.

Decompositie van stappen

In PROforma kunnen groepen taken gemodelleerd worden als onderdeel van een *plan*. Een plan is beëindigd wanneer alle niet-optionele deeltaken beëindigd zijn. De optionele deeltaken worden dan in dat geval ook beëindigd.

Procedurele kennis

- Twee PROforma-taken kunnen *sequentieel* worden uitgevoerd d.m.v. een scheduling constraint tussen beide taken.
- *Parallele* taken worden gemodelleerd door scheduling constraints tussen deze taken en een voorgaande taak te plaatsen. In een plan zijn de taken ook parallel wanneer deze als eerste starten, d.w.z. wanneer er geen taak aan vooraf gaat.

- Wanneer parallelle taken als laatste in een plan gestart worden is *synchronisatie* gewaarborgd, omdat eventuele taken na het plan pas starten wanneer de parallelle taken voltooid zijn. Synchronisatie van parallelle taken die niet als laatste in een plan starten wordt gerealiseerd door scheduling constraints tussen deze taken en een volgende taak te plaatsen.
- Voor het representeren van taken die *tegelijk moeten eindigen* zijn geen expliciete constructies beschikbaar. Een mogelijkheid is om de taken parallel uit te voeren, waarbij in de termination-conditions van beide taken (in dit geval plannen) aan de activatie van de andere taak gerefereerd wordt. Voorbeeld: de termination-condition van een plan B is als volgt gespecificeerd: `end(plan A) isbefore now`.
- In PROforma kunnen *cyclische* taken worden gemodelleerd. Het aantal cycli kan hierbij expliciet of middels een conditie worden gerepresenteerd. Ook kan worden weergegeven hoeveel tijd er tussen twee opeenvolgende cycli moet zitten.
- Net als bij taken die tegelijk moeten worden uitgevoerd kent PROforma ook voor het *conditioneel vertakken* geen expliciete constructies, zoals uit stroomdiagrammen. De vertakkingen zullen gemodelleerd moeten worden als parallelle taken, waarin in de precondities wordt besloten of de taak uitgevoerd moet worden. Er is op deze manier dus geen sprake van een enkelvoudige beslissing: bij het uitvoeren van een protocol moet voor elke tak apart worden besloten of deze geactiveerd wordt. Een enkelvoudige beslissing kan eventueel gesimuleerd worden door de takken als kandidaten van een *decision* te laten optreden. Dit neemt echter niet weg dat in de precondities alsnog het resultaat van de decision moet worden opgenomen.
- Voor elke taak is er een optie *optional* waarmee aangegeven kan worden dat de taak optioneel is. Zo'n taak is echter slechts echt optioneel wanneer het niet gevolgd wordt door een andere, niet-optionele, taak; wanneer dit wel het geval is zal de taak alsnog uitgevoerd moeten worden om door te gaan (zie figuur 6.1). Hierbij dient opgemerkt te worden dat een plan beëindigd is wanneer alle niet-optionele taken beëindigd zijn. Om in zo'n geval een succesvolle uitvoering van de optionele taak te garanderen zal deze voltooid moeten zijn voordat de niet-optionele taken beëindigd zijn (zie voorbeeld in figuur 6.1).

Precondities

In alle taken van een protocol kunnen *precondities* worden opgenomen. Wanneer een taak voor activatie in aanmerking komt en de preconditie op dat moment (nog) niet geldt wordt de taak afgewezen. Eventuele taken die volgen op deze taak worden dan ook niet uitgevoerd.

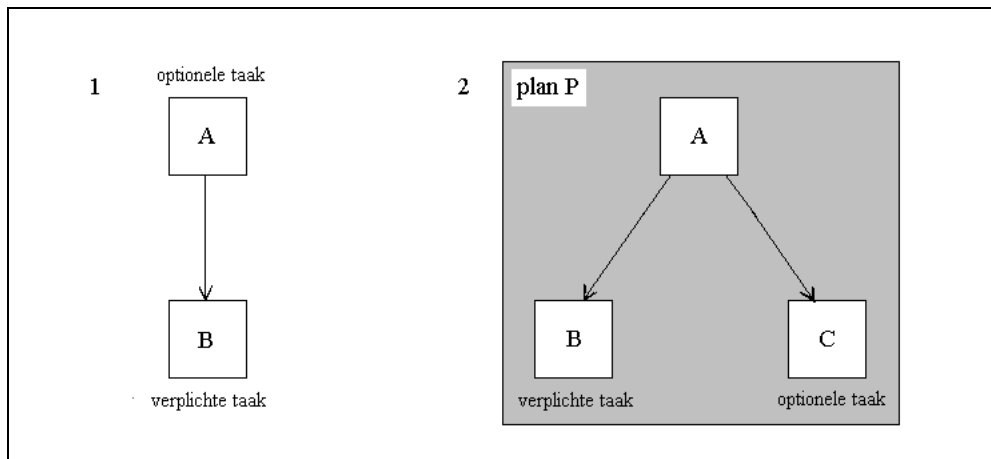
Voortijdig beëindigen van stappen

Een taak wordt *afgebroken* wanneer er aan de *abort-condition* is voldaan (in het geval van een plan) of wanneer het deel uitmaakt van een afgebroken plan. Eventuele taken volgend op een afgebroken taak worden niet uitgevoerd. Het plan waar de afgebroken taak deel van uitmaakt wordt echter niet afgebroken. Om dit toch te realiseren zou er in de abort-condition van het plan eventueel naar de toestanden van deeltaken verwezen kunnen worden.

Wanneer de *termination-condition* van een plan geldt wordt het plan direct (succesvol) beëindigd. De deeltaken van dat plan worden in dat geval ook beëindigd.

Postcondities

Met het attribuut *postcondition* kan, middels een variabele-waarde assignment een postconditie gespecificeerd worden.



Figuur 6.1: Optionele taken. In figuur 1 volgt een verplichte taak B op een optionele taak A. Om activatie van taak B te realiseren zullen alle daar aan voorafgaande taken uitgevoerd moeten zijn, dus ook taak A. Taak A is hier dus slechts in naam optioneel, in gedrag is het verplicht geworden. In figuur 2 worden de verplichte taak B en optionele taak C tegelijk geactiveerd. Wanneer taak B voltooid is is het gehele plan P beëindigd, en wordt taak C dus niet geheel uitgevoerd. Om te realiseren dat taak C toch geheel wordt uitgevoerd zal deze voltooid moeten zijn voordat taak B voltooid is.

Temporele kennis

PROforma houdt de tijdstippen bij van toestandsovergangen en bekend worden van data-items. In expressies, zoals boolse expressies, kan er naar deze tijdstippen verwezen worden, waarbij representatie van tijd in verschillende eenheden (*days*, *months*, enz.) mogelijk is. Op deze manier kan bijvoorbeeld in de termination-condition de maximale duur van een plan worden gespecificeerd. Het is niet mogelijk om een taak vertraagd te laten starten. Daarnaast kan temporele onzekerheid omtrent begin en einde van intervallen niet worden uitgedrukt.

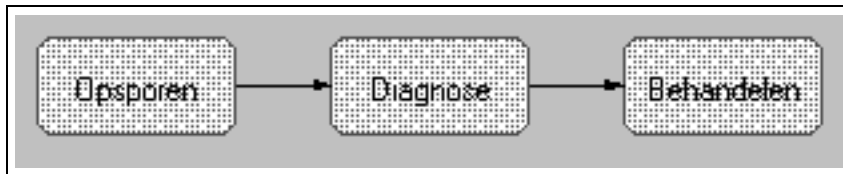
Representatie van data

Er zijn in PROforma drie manieren waarop gegevens verkregen kunnen worden: door de waarde van een data-item aan de gebruiker te vragen, door de waarde van een data-item af te leiden middels het attribuut *derivation rule* of door tot een kandidaat te concluderen in een *decision*. Elk gegeven is hier gemodelleerd als een globale variabele met datatypen zoals integers, booleans, verzamelingen of tijdstippen. Tijdens het uitvoeren van een protocol kan een variabele over de tijd meerdere waarden aannemen; er zijn functies beschikbaar waarmee de juiste waarde uit zulke waarden geselecteerd kan worden. De eenheid van de waarde van een variabele kan worden vastgelegd.

6.2 Het PROforma-model van de cholesterolrichtlijn

plan Cholesterol

Op het hoogste niveau bestaat het model uit drie plannen: **Opsporen**, **Diagnose** en **Behandelen**. Deze worden ook in die volgorde uitgevoerd (zie figuur 6.2 voor een grafische representatie en figuur 6.3 voor de code van het plan Cholesterol). De volgorde van en de samenhang tussen deze plannen weerspiegelt de gang van een patiënt van anamnese naar een eventuele therapie. In het plan **Opsporen** wordt patiënt-data opgevraagd en worden de toegangscriteria



Figuur 6.2: De drie plannen van het cholesterol-protocol.

```

plan :: cholesterol ;
  caption :: 'Cholesterol' ;
  description :: 'Cholesterol-richtlijn' ;
  component :: opsporen ;
  component :: diagnose ;
    schedule_constraint :: completed(opsporen) ;
  component :: behandelen ;
    schedule_constraint :: completed(diagnose) ;
end plan .
  
```

Figuur 6.3: De attributen behorende bij plan Cholesterol uit figuur 6.2.

voor de richtlijn geformuleerd. Deze toegangscriteria zijn opgenomen in de preconditionie van plan **Diagnose**. In dit plan is het lichamelijk onderzoek, de interpretatie van het lichamelijk onderzoek, het bepalen van het 10-jaarsrisico en de therapie-beslissing gemodelleerd. De therapie-beslissing is weer opgenomen in de preconditionie van plan **Behandelen**, waarin statines worden voorgeschreven en de behandeling wordt geëvalueerd. Het protocol is beëindigd wanneer het plan **Behandelen** voltooid is, of wanneer voor aanvang van de plannen **Diagnose** of **Behandelen** de respectievelijke preconditionies niet gelden. De drie plannen zullen hieronder afzonderlijk in detail worden besproken.

plan Opsporen

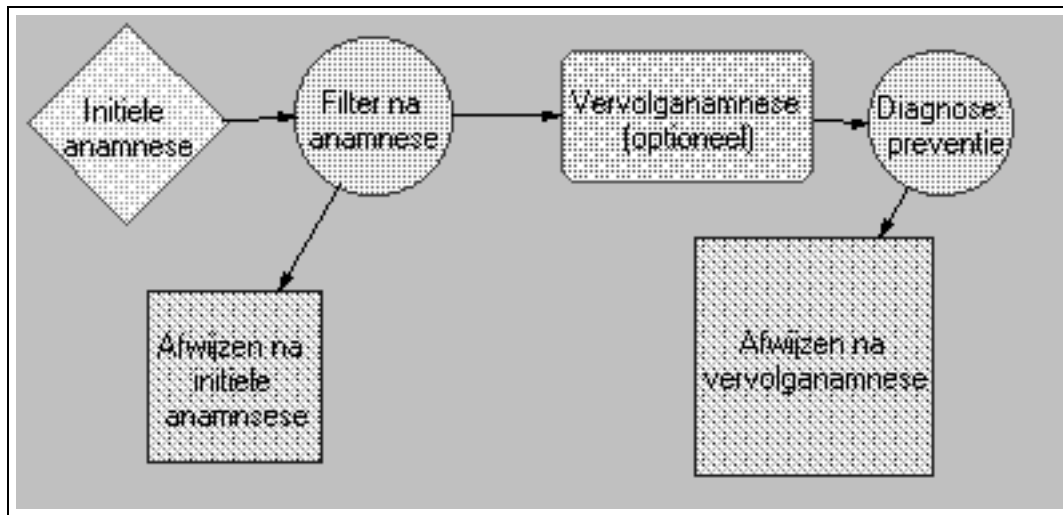
Voor de grafische representatie: zie figuur 6.4. In sommige gevallen komt een patiënt niet in aanmerking voor behandeling met statines, bijvoorbeeld bij een te hoge leeftijd of de aanwezigheid van een ernstige ziekte. De informatie voor deze beslissing wordt opgevraagd met de enquiry **Initiele anamnese**, en aansluitend wordt in **Keuze:na anamnese** besloten of de patiënt de richtlijn vervolgt.

Vervolgens wordt er in plan **Vervolganamnese** aanvullende informatie opgevraagd, zoals diabetes mellitus en roken. In de *decision* **Diagnose:preventie** tenslotte wordt de patiënt geclassificeerd in één van de volgende groepen **primaire preventie**, **secundaire preventie** en **afgewezen**.

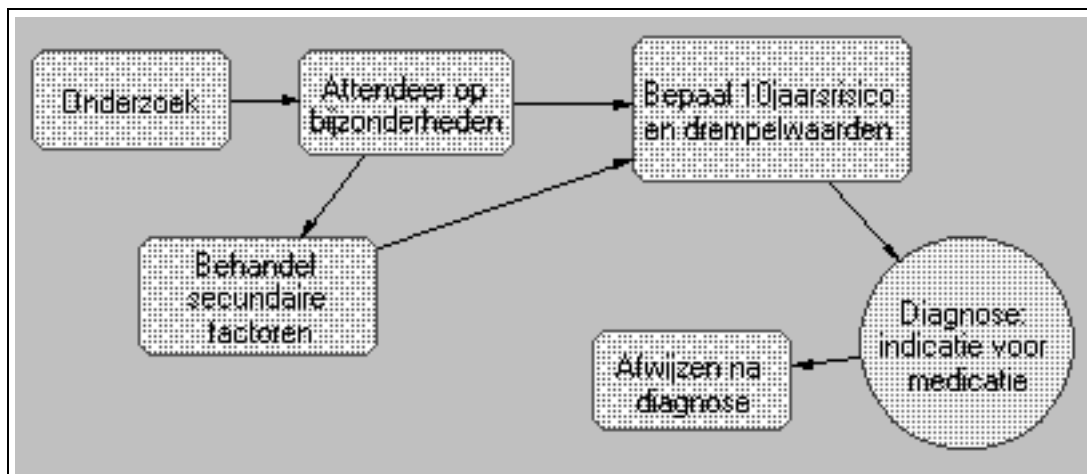
De plannen **Afwijzen na initiële anamnese** en **Afwijzen na vervolganamnese** worden actief wanneer de patiënt niet in aanmerking komt voor medicatie, en laat de reden van afwijzing zien.

plan Diagnose

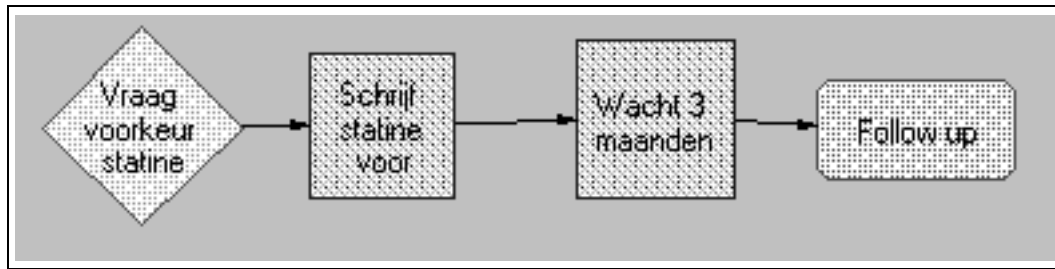
Voor de grafische weergaven van dit plan: zie figuur 6.5. Dit plan start met het plan **Onderzoek** waarin parallel het bloedonderzoek, het onderzoek naar erfelijke factoren en het onderzoek naar secundaire oorzaken uitgevoerd wordt. Vervolgens wordt het plan **Attendeer op bijzonderheden** uitgevoerd waarin de gebruiker op eventuele bijzonderheden uit het lichamelijk onderzoek wordt geattendeerd. Afhankelijk van de aanwezigheid van secundaire oorzaken wordt ofwel direct, ofwel na behandeling van de secundaire oorzaken het 10-jaarsrisico bepaald (dit is gemodelleerd m.b.v. preconditionies).



Figuur 6.4: De taken van het plan Opsporen.



Figuur 6.5: De taken van het plan Diagnose.



Figuur 6.6: De taken van het plan *Behandeling*.

Omdat *PROforma* geen lijsten kent gaf het modelleren van de risicotabel een probleem. Zonder lijsten bleek het niet mogelijk om de tabel en het zoeken in de tabel gescheiden te modelleren. Dit resulteerde in een omslachtige oplossing waarin de tabel opgeplitst is in 16 deeltabellen, voor alle permutaties van de binaire data-items **geslacht**, **hypertensie**, **roken** en **diabetes**. Het zoeken gebeurt dan in twee fasen: allereerst wordt bepaald in welke deeltabel gezocht moet worden, om vervolgens uit de juiste deeltabel de vier risico's te halen (omdat het exacte risico volgt uit interpolatie van de risico's voor de vier (naar boven resp. naar beneden) afgeronde waarden voor **leeftijd** en **ratio**). Het zoeken naar de vier risico's werd gerealiseerd door een plan vier keer uit te voeren, en waarbij m.b.v. data-item **teller** wordt bijgehouden naar welke risico gezocht moet worden. De teller krijgt z'n, willekeurige, waarde middels een derivation rule. Bij elke cyclus krijgt het data-item teller er een waarde bij. Uit het aantal waarden voor data-item teller is dan af te leiden in welke cyclus het protocol zich bevindt. Deze omslachtige methode was noodzakelijk omdat *PROforma* in een protocol geen statements, zoals die voorkomen in imperatieve programmeertalen, toelaat. Een statement als `teller = teller + 1` was hier dan ook niet mogelijk.

Decision Diagnose:indicatie voor medicatie beslist vervolgens of de patiënt in aanmerking komt voor medicatie. De actie **Afwijzen na diagnose** wordt actief wanneer de patiënt niet in aanmerking komt voor medicatie, en laat de reden van afwijzing zien.

plan *Behandelen*

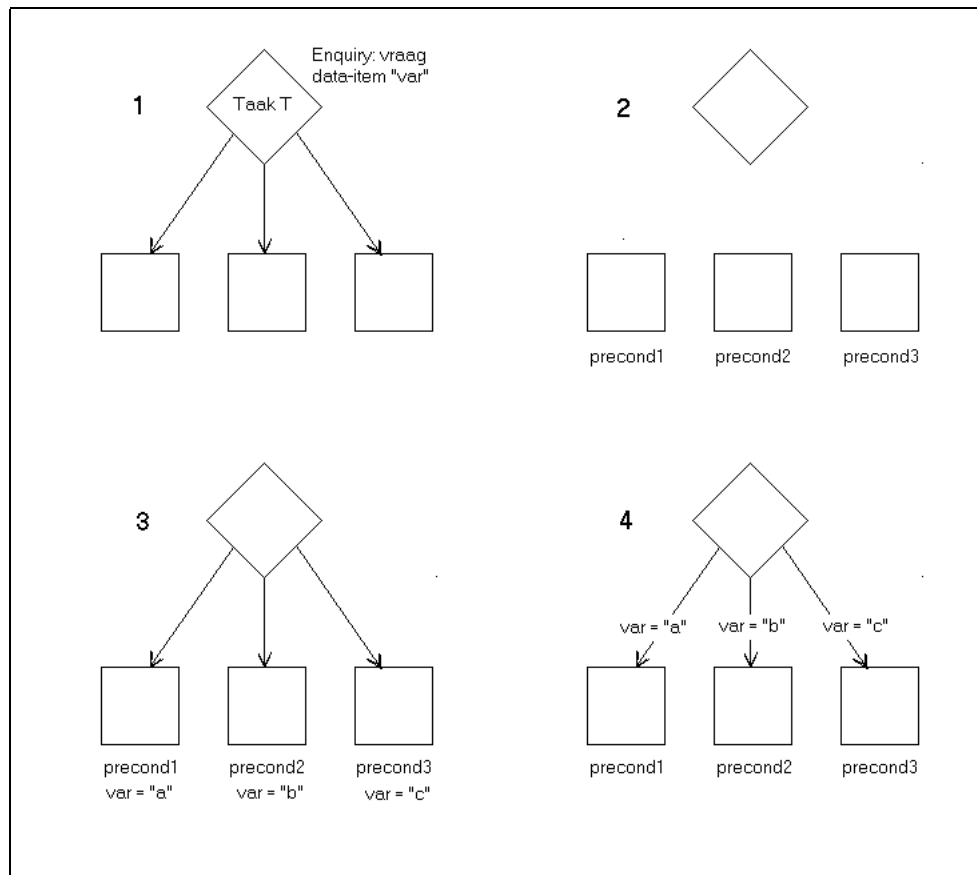
Het doel van dit plan (zie ook figuur 6.6) is het voorschrijven van medicatie, het eventueel wijzigen van de dosis na drie maanden en het jaarlijks controleren van de patiënt op eventuele wijzigingen in de klinische toestand. Na het voorschrijven van medicatie zal dus drie maanden gewacht moeten worden voordat plan *Follow-up* start. De taken van *PROforma* kennen geen attributen waarin dit direct kan worden uitgedrukt. Gekozen is daarom voor een *action* die aangeeft dat de arts drie maanden moet wachten.

6.3 Bevindingen tijdens het modelleren in *PROforma*

Deze paragraaf beschrijft enkele ervaringen en aanbevelingen die specifiek betrekking hebben op het modelleren in *PROforma*. Aspecten van *PROforma* die ook Asbru betreffen zullen in hoofdstuk 8 besproken worden.

Relatie tussen constraintpijlen en preconditionies

Het modelleren in *PROforma* levert problemen op wanneer er na een bepaalde taak T uit meerdere mogelijke opvolgers (een opvolger is in deze context een taak die direct volgt op taak T) slechts één taak wordt gekozen (zie eerste figuur in figuur 6.7):



Figuur 6.7: Precondities en constraintpijlen.

- Alle opvolgers van de taak komen in aanmerking voor activatie; voor elke opvolger zal dan ook in de desbetreffende preconditionie de voorwaarden moeten zijn opgenomen wanneer voor de opvolger te kiezen. Voor een opvolger kan echter reeds een preconditionie geformuleerd zijn (zie tweede figuur in figuur 6.7), bijvoorbeeld een 'safety-condition'. Na toevoeging van de constraintpijl-afhankelijke, 'procedurele' of 'besturings', voorwaarden is het niet mogelijk de 'oorspronkelijke' voorwaarden van de 'besturings' voorwaarden te onderscheiden (zie derde figuur in figuur 6.8). De preconditionie van een taak heeft dus een dubbelzinnige betekenis, wat verwarrend kan werken. Daarnaast wordt bij het kopiëren van een taak behalve de 'oorspronkelijke' preconditionie ook de 'procedurele' preconditionie overgenomen.
- In dit geval bestaat er ook onduidelijkheid hoe de uitgaande constraintpijlen van taak T te interpreteren. Al deze pijlen worden gevolgd. In dit geval wordt er echter uiteindelijk slechts één van de opvolgende taken uitgevoerd. Dit wordt grafisch niet duidelijk.

Dit probleem zou opgelost kunnen worden wanneer het mogelijk zou zijn ook voor constraintpijlen preconditionies te formuleren (zie vierde figuur in figuur 6.7). Op deze manier bestaan de preconditionies van taken uit 'oorspronkelijke' voorwaarden, en de preconditionies van constraintpijlen uit 'besturings' voorwaarden. Eventueel kan een protocol-editor automatisch alle uitgaande pijlen van een taak markeren/kleuren wanneer de constraint-preconditionies elkaar uitsluiten.

Dubbelzinnige interpretatie van taak *decision*

De relatie tussen een *decision* en de taken die volgen op een *decision* is wat verwarrend. Constraintpijlen drukken in PROforma de volgorde van de taken uit, dus ook pijlen die volgen op een *decision*. De beslissing welke van de opvolgers van een *decision* geactiveerd worden (zoals dit gespecificeerd is in de precondities van de taken) is dan ook niet per se afhankelijk van het resultaat van de *decision*. Intuïtief (vanwege gelijkenis met gangbare diagrammen zoals stroomdiagrammen) bestaat echter de neiging om in sommige gevallen de uitgaande pijlen van een *decision* te interpreteren als verschillende opties waarvan er één wordt gevolgd, en waarbij het afhankelijk is van het resultaat van de *decision* welke wordt gevolgd.

Bij het modelleren van de cholesterolrichtlijn is dit probleem opgelost door twee ‘klassen’ van de taak *decision* te introduceren, nl. *Diagnose* en *Keuze*, en deze klassen te definiëren door naamgeving en het gebruik van deze *decisions*:

- Bij een *diagnose*-*decision* staat er “Diagnose:” in de naam. Deze wordt gebruikt wanneer er bepaalde kennis moet worden afgeleid. In de precondities van de taken die volgen op uitgaande pijlen van een *diagnose* wordt er niet per se gerefereerd aan de kandidaten van de *decision*. De constraintpijlen moeten dus geïnterpreteerd worden als ‘volgorde’ -pijlen.
- Bij een *keuze*-*decision* staat er “Keuze:” in de naam. Deze wordt gebruikt wanneer er besloten moet worden wat de volgende taak zal zijn. De precondities van taken die volgen op uitgaande pijlen van een *keuze* refereren hier dus wel aan de kandidaten in de *decision*. De constraintpijlen moeten dus geïnterpreteerd worden als ‘keuze’ -pijlen.

Inferentie van kennis

PROforma kent twee manieren om kennis te infereren, nl. middels een *decision* en door middel van een *derivation-rule* van een *data-item*. In beide gevallen zal er een taak moeten worden uitgevoerd om de kennis te infereren: de *derivation-rule* wordt pas geactiveerd wanneer er in een enquiry het *data-item* opgevraagd wordt. Het is dus niet mogelijk om kennis af te leiden zonder expliciet gebruik te maken van een taak.

Triggering

Het triggeren van een taak moet actief door de gebruiker gebeuren, door tijdens toepassing van het protocol expliciet aan te geven dat de triggerconditie van de taak geldt. Dit gaat dus niet automatisch. Het blijkt hierbij ook niet van belang te zijn of de triggerconditie werkelijk geldt. Het automatisch starten van een taak waarvan de triggerconditie geldt (en alleen dan!) is echter wel wat je onder triggering mag verstaan.

Wijzigen van gegevens

In PROforma bleek het niet mogelijk (althans met de tools waarmee gewerkt werd) om tijdens het uitvoeren van het protocol gegevens te wijzigen. Dit kan lastig zijn als bijvoorbeeld blijkt dat gegevens foutief zijn ingevoerd.

Hoofdstuk 7

Modelleren van de cholesterolrichtlijn in Asbru

Voor Asbru was er geen modelleersoftware beschikbaar. Het protocol is geschreven in een Word-document.

7.1 Representatie van de informatietypen in medische richtlijnen in Asbru

Doel

In Asbru wordt een doel gerepresenteerd m.b.v. intentions. Een intention beschrijft een doel, in een streng formaat, als een toestand of een handeling die bereikt, gehandhaafd of vermeden moeten worden. Deze intenties hebben invloed op de uitvoering van een protocol in de zin dat ze gebruikt worden bij het selecteren van plannen.

Atomaire stap

Asbru kent twee basisconstructies waarmee eenvoudige acties van een arts gemodelleerd kunnen worden. Met element *ask* kunnen patiëntgegevens worden verzameld, en met het element *user-performed* kunnen ‘externe’ handelingen worden gemodelleerd.

Decompositie van stappen

De *plan-body* van een plan specificeert de stappen die moeten worden uitgevoerd. Deze stappen omvatten ondermeer atomaire acties, variabele-assignments en if-then-else-statements. Daarnaast kunnen er in zo’n plan-body ook weer deelplannen voorkomen zodat een decompositie van stappen in Asbru gerealiseerd kan worden. Het plan is beëindigd wanneer alle verplichte stappen in de plan-body van het plan voltooid zijn, en er bovendien aan de complete-condition en time-annotation voldaan is.

Procedurele kennis

- Een groep stappen wordt in Asbru gemodelleerd in het element *subplans* van de plan-body. De volgorde van de stappen is afhankelijk van het subplans-attriboot *type*. Bij het modelleren van *sequentiële* stappen is het element subplans van het type *sequential*. De volgorde waarop de stappen in het element subplans zijn opgesomd is dan ook de volgorde waarin de stappen worden uitgevoerd.
- Bij het modelleren van *parallelle* stappen is het element subplans van het type *parallel*. Het modelleren van stappen die parallel mogen worden uitgevoerd maar die niet per se tegelijkertijd hoeven te starten wordt gerealiseerd door gebruik van subplans van het type *unordered*.

- *Synchronisatie* is in Asbru gewaarborgd doordat:
 - een plan is beëindigd wanneer de stappen in de plan-body zijn beëindigd (afgezien van de time-annotation en andere constraints). Wanneer deze stappen parallel zijn gemodelleerd zijn de stappen *synchroon*.
 - in een reeks sequentiële stappen een stap pas wordt geactiveerd wanneer de vorige stap is voltooid. Wanneer deze laatstgenoemde stap een reeks parallelle stappen betreft wordt er dus gewacht tot de parallelle stappen voltooid zijn.
- Asbru kent geen expliciete constructies voor het representeren van stappen die *tegelijk moeten eindigen*. Een mogelijkheid is om de stappen (in dit geval plannen) parallel uit te voeren, waarbij in de complete-conditions van beide plannen aan de toestand van het andere plan gerefereerd wordt. De complete-conditie van de ene plan geldt dan wanneer de toestand van de andere plan **completed** is. Om direct te eindigen bij het waar worden van de complete-condition zal een plan geen verplichte stappen mogen bevatten.
- In Asbru kunnen stappen *cyclisch* worden uitgevoerd. Met *cyclical-complete-conditions* wordt gespecificeerd wanneer een cyclische stap beëindigd is. Dit worden aangegeven door een bepaald tijdstip, het aantal keren dat het plan uitgevoerd moet worden of met een ‘until’-conditie. Daarnaast kan met *any-repeat-specification* worden aangegeven wanneer en met welke tussenpozen de cyclisch weerkerende stappen moeten starten.
- Het *conditioneel vertakken* kan in Asbru op twee manieren gerealiseerd worden. Expliciet, door in de plan-body gebruik te maken van een *if-then-else-statement*. Impliciet, door de vertakkingen te modelleren als parallelle plannen en in de *filter-conditions* van de plannen de voorwaarden op te nemen wanneer de plannen te starten.
- Voor een groep stappen kan in de *continuation specification* worden gespecificeerd welke stappen optioneel zijn. Met de *activate mode* kan vervolgens worden aangegeven dat een plan pas start na goedkeuring van de gebruiker. In dit geval moet de gebruiker binnen een bepaalde tijd reageren, anders wordt het plan niet uitgevoerd.

Precondities

Asbru kent feitelijk twee soorten precondities, nl de *filter-* en de *setup-precondition*. Het verschil tussen beide precondities wordt duidelijk wanneer een plan in aanmerking komt voor activatie: een plan waarvan de filter-preconditie niet geldt wordt afgewezen, terwijl een setup-preconditie de tijd krijgt (deze tijd is gedefinieerd in het attribuut *waiting period*) om vervuld te worden. Wanneer de setup-preconditie niet binnen deze waiting period vervuld wordt, wordt het desbetreffende plan afgewezen.

Voortijdig beëindigen van stappen

Een plan dat reeds actief is wordt *afgebroken* wanneer er aan de *abort-condition* is voldaan, wanneer de uitvoering van een plan vanwege een time-annotation stopt terwijl de verplichte stappen nog niet zijn voltooid, of wanneer één van de verplichte stappen wordt afgebroken of afgewezen wegens een falende filterconditie.

Daarnaast is het mogelijk om een plan succesvol te *beëindigen* terwijl niet alle plannen in de plan-body voltooid zijn. Dit wordt gerealiseerd door voor de plannen in de plan-body onderscheid te maken tussen verplichte en optionele plannen (met een continuation specification) en in de *complete-condition* de voorwaarden te specificeren wanneer het plan beëindigd is. Een plan is dan beëindigd wanneer de complete-conditie geldt en de verplichte plannen voltooid zijn.

Postcondities

Een postconditie van een plan wordt in Asbru gerepresenteerd door het (kwalitatieve) effect van het plan op een bepaalde parameter te representeren.

Temporele kennis

Asbru biedt m.b.v. *time-annotations* de mogelijkheid om het begin- en eindtijdstip alsmede de duur van een plan weer te geven. Deze tijd-specificaties hoeven niet exact te zijn: het is mogelijk om de onzekerheid over de specificaties weer te geven. De duur van een plan kan bijvoorbeeld worden beschreven met een minimum en een maximum duur. Met deze *time-annotations* is het ook mogelijk om temporele onzekerheid omtrent patiëntgegevens weer te geven. De begin- en eindtijdstippen zijn relatief t.o.v. een referentie-tijdstip. Deze referentie-tijdstip kan refereren aan toestandsovergangen van een plan of aan een willekeurige expressie die bij evaluatie een tijdstip oplevert. Het tijdstip van bekend worden van een parameter of een variabele wordt in Asbru wel bijgehouden, maar het is niet mogelijk om aan dit tijdstip te refereren. Ook Asbru kent verschillende eenheden voor tijd.

Representatie van data

De gegevens die aan de gebruiker worden gevraagd of die via een input-kanaal tot het systeem komen worden opgeslagen als *parameters*. Met parameters kunnen ook indirecte gegevens worden gerepresenteerd zoals het resultaat van een rekenkundige bewerking, een klassificatie van een kwantitatieve parameter of het gemiddelde of de verandering van een parameter-waarde. Dit laatste is mogelijk omdat een parameter in de loop van de tijd meerdere waarden kan krijgen; al deze waarden worden, gekoppeld aan het tijdstip waarop de waarde bekend werd, opgeslagen. In *parameter-propositions* is het mogelijk te refereren aan de waarde van een parameter op een bepaald tijdstip.

Een andere manier om gegevens te representeren is met een *variabele*. Deze krijgt zijn waarde toegewezen in de variabele-definitie of middels een variabele-assignment in de plan-body van een plan; in een variabele kunnen dus geen externe gegevens, zoals opvraagbare data, opgeslagen worden. Bovendien wordt, in tegenstelling tot parameters, van een variabele slechts de laatste waarde bewaard.

7.2 Het Asbru-model van de cholesterolrichtlijn

In de handleiding van Asbru werd de taal gepresenteerd in de XML-syntax. Het modelleren van de richtlijn in dit formaat zou een *plan-library* opleveren, gerepresenteerd als een boomstructuur met diepe vertakkingen. Terwille van de overzichtelijkheid is echter gekozen voor een meer ‘plat’ formaat, zonder informatie te verliezen. In dit formaat worden eerst de plannen beschreven en daarna andere elementen zoals parameter-definities.

In het onderstaande zal het cholesterolprotocol in Asbru besproken worden aan de hand van de plannen **Cholesterol**, **Opsporen**, **Diagnose** en **Behandeling**.

plan Cholesterol

Het protocol start tijdens het uitvoeren met het plan **Cholesterol**. In dit plan worden drie plannen sequentieel uitgevoerd, nl. **Opsporen**, **Diagnose** en **Behandelen** (zie figuur 7.1). In het plan **Opsporen** (zie figuur 7.2) worden patiëntgegevens opgevraagd en de toegangscriteria voor de richtlijn geformuleerd. Deze toegangscriteria zijn opgenomen in de filterconditie van

PLAN	CHOLESTEROL
Variables	preventie type : preventie-schaal hypertensie, indicatie-voor-behandeling type : boolean
Intentions	ACHIEVE OVERALL STATE: known(indicatie-voor-behandeling)
Plan Body	SEQUENTIAL wait-for all Opsporen Diagnose Behandelen

Figuur 7.1: Het plan Cholesterol

PLAN	OPSPOREN				
Intentions	ACHIEVE OVERALL STATE: known(preventie)				
Conditions	<table border="1"> <tr> <td>Complete</td><td>preventie = "primaire" or preventie = "secundaire"</td></tr> <tr> <td>Abort</td><td> <pre>(levensverwachting>5jr = "nee" OR verminderde- linkerkamerfunctie = "ja" OR manifest-hartfalen = "ja" OR ernstige-ziekte = "ja OR (leeftijd >="70" AND geslacht = "man") OR (leeftijd >="75" AND geslacht = "vrouw"))</pre> Explanation slot: Display "Patiënt is afgewezen wegens te hoge leeftijd of een bepaalde ziekte" OR (preventie = "geen") Explanation slot: Display "Geen HVZ en er zijn geen risicofactoren gevonden" </td></tr> </table>	Complete	preventie = "primaire" or preventie = "secundaire"	Abort	<pre>(levensverwachting>5jr = "nee" OR verminderde- linkerkamerfunctie = "ja" OR manifest-hartfalen = "ja" OR ernstige-ziekte = "ja OR (leeftijd >="70" AND geslacht = "man") OR (leeftijd >="75" AND geslacht = "vrouw"))</pre> Explanation slot: Display "Patiënt is afgewezen wegens te hoge leeftijd of een bepaalde ziekte" OR (preventie = "geen") Explanation slot: Display "Geen HVZ en er zijn geen risicofactoren gevonden"
Complete	preventie = "primaire" or preventie = "secundaire"				
Abort	<pre>(levensverwachting>5jr = "nee" OR verminderde- linkerkamerfunctie = "ja" OR manifest-hartfalen = "ja" OR ernstige-ziekte = "ja OR (leeftijd >="70" AND geslacht = "man") OR (leeftijd >="75" AND geslacht = "vrouw"))</pre> Explanation slot: Display "Patiënt is afgewezen wegens te hoge leeftijd of een bepaalde ziekte" OR (preventie = "geen") Explanation slot: Display "Geen HVZ en er zijn geen risicofactoren gevonden"				
Plan Body	SEQUENTIAL wait-for none ANY-ORDER wait-for all ask "leeftijd", ask "geslacht", ask "hvz", ask "levensverwachting>5jr", ask "verminderde-linkerkamerfunctie", ask "ernstige-ziekte", ask "manifest-hartfalen" Aanvullende-anamnese if hvz = "ja" then preventie := "secundaire" else if (hypertensie = "ja" or chv-in-familie = "ja" or diabetes_mellitus = "ja") then preventie := "primaire" else preventie := "geen"				

Figuur 7.2: Het plan Opsporen

plan Diagnose (zie figuur 7.3). In dit plan is het lichamelijk onderzoek, de interpretatie van het lichamelijk onderzoek, het bepalen van het 10-jaarsrisico en de therapie-beslissing gemodelleerd. De therapie-beslissing is weer opgenomen in de filterconditie van plan Behandelen (zie figuur 7.4), waarin statines worden voorgeschreven en de behandeling wordt geëvalueerd. De drie deelplannen in plan Cholesterol moeten alle worden uitgevoerd; dit is gespecificeerd in de continuation specification (het **wait-for-element**). De plannen Opsporen en Diagnose worden afgebroken wanneer blijkt dat de patiënt niet in aanmerking komt voor behandeling met statines. In dat geval wordt, in element **explanation slot** van de abort conditions, uitgelegd waarom de desbetreffende plannen zijn afgebroken. Het plan is vervolgens voltooid wanneer het plan Behandelen voltooid is. Waar mogelijk zijn de variabelen zo lokaal mogelijk gedefinieerd. In plan Cholesterol zijn variabelen gedefinieerd die globaal zijn t.o.v. het cholesterolprotocol, te weten **preventie**, **hypertensie**, en **indicatie-voor-behandeling**.

plan Opsporen

In sommige gevallen komt een patiënt niet in aanmerking voor behandeling met statines, bijvoorbeeld bij een te hoge leeftijd of de aanwezigheid van een ernstige ziekte; in dat geval

PLAN	DIAGNOSE
Variables	risico type : amount unit : %
Intentions	ACHIEVE OVERALL STATE: known(indicatie-voor-behandeling)
Conditions	Filter preventie = "primair" or preventie = "secundair"
	Complete is-known(indicatie-voor-behandeling)
Plan Body	SEQUENTIAL wait-for Lichamelijk-onderzoek Beslissing-behandeling abort-if Behandel-secundaire-oorzaken : aborted wait-for-optional-plans = "yes" Lichamelijk-onderzoek ask "erfelijke-factoren" ask "secundaire-oorzaken" Attendeer-gebruiker Behandel-secundaire-oorzaken Bepaal-risico Beslissing-behandeling

Figuur 7.3: Het plan Diagnose

wordt het plan afgebroken. Met de 'ask'-elementen wordt de informatie voor deze beslissing opgevraagd. In het plan **Aanvullende-anamnese** worden de risicofactoren **diabetes-mellitus**, **roken**, **chz-in-familie** en **behandeld-voor-hypertensie** gevraagd. Wanneer de patiënt niet voor hypertensie wordt behandeld wordt uit de bloeddrukgegevens bepaald of de patiënt hypertensief is.

Het plan **Opsporen** eindigt met het vaststellen (m.b.v. een if-statement) of het in het geval van de patiënt om primaire, secundaire of geen preventie gaat. Indien het geen preventie betreft wordt het plan afgebroken.

plan Diagnose

Het plan start met het bloedonderzoek, een onderzoek naar erfelijke factoren en een onderzoek naar secundaire oorzaken. De onderzoeken worden tegelijkertijd gestart en zijn bevat in het plan **Lichamelijk-onderzoek**.

Het plan **Attendeer-gebruiker** is optioneel (het komt niet voor in de continuation specification): het wordt alleen geactiveerd bij hoge cholesterolconcentraties of wanneer er erfelijke factoren zijn gevonden (dit is gespecificeerd in de filter-conditie van **Attendeer-gebruiker**). In dat geval wordt de gebruiker dan geattendeerd op de bijzondere bevindingen.

Ook het daarop volgende plan **Behandel-secundaire-oorzaken** is optioneel, en heeft de aanwezigheid van secundaire oorzaken als filter-conditie. Dit plan behandelt de secundaire oorzaken m.b.v. een **user-performed** actie. Met het Asbru-element **user-performed** worden externe, door de gebruiker uit te voeren, handelingen gemodelleerd. De gebruiker krijgt te zien welke handeling moeten worden gedaan, en hij/zij dient vervolgens zelf aan te geven of de handeling voltooid is. Na afloop wordt de gebruiker gevraagd of het protocol voortgezet dient te worden. Indien dit het geval is worden de nieuwe cholesterolwaarden die gebruikt worden bij het bepalen van het 10-jaarsrisico opgevraagd. Indien dit niet het geval is wordt het plan, en bijgevolg plan **Diagnose**, afgebroken.

Het modelleren van de risicotabel in het plan **Bepaal-risico** gaf problemen omdat Asbru slechts 1-dimensionale lijsten kent. Dit werd opgelost door de 8-dimensionale tabel af te beelden op een lijst, genaamd **risico-lijst**. De eerste helft van de lijst bestaat hierbij

PLAN	BEHANDELEN	
Conditions	Filter	indicatie-voor-behandeling = "ja"
Plan Body	SEQUENTIAL wait-for all ask "voorkeur-statine" user-performed "Schrijf de gekozen statines voor (20-40 mg/dag)" plan-activation time-annotation time-range starting-shift earliest plan-state-transition(first-considered, self) + 3 month Follow-up user-performed "Voer jaarlijkse controle uit"	

Figuur 7.4: Het plan Behandelen

uit de risicowaarden van de man en de tweede helft uit risicowaarden van de vrouw. Van de eerste helft van de lijst is vervolgens de eerste helft weer gereserveerd voor mannelijke diabetici en de tweede helft voor mannen zonder diabetes mellitus, enz. Op deze manier zijn de risicowaarden voor alle in de risicotabel voorkomende combinaties van geslacht, diabetes mellitus, roken, hypertensie, leeftijd en ratio in de lijst gerepresenteerd. Voor het zoeken in de lijst worden eerst de relatieve posities voor de actuele waarden van geslacht, diabetes mellitus, roken en hypertensie bepaald, middels if-statements, en opgeteld. Het resultaat **geslacht-diabetes-roken-hypertensie-positie**, en de relatieve posities van de afgeronde waarden voor leeftijd en ratio (deze worden middels data-abstractie bepaald) wordt vervolgens gebruikt voor het berekenen van de vier posities waarmee de benodigde risicowaarden te vinden zijn. Aan de hand van de vier posities worden met de functie **get-element-on-position** de vier benodigde risicowaarden uit de lijst gelezen. Uit de vier risicowaarden wordt vervolgens het risico berekend.

Het plan eindigt met uitvoering van plan **Beslissing-behandeling**. In dit plan wordt met plan **Bepaal-drempel-en-uitzondering** de drempelwaarde en de uitzondering geformuleerd. Vervolgens wordt met een if-statement beslist of de patiënt statines voorgeschreven krijgt.

plan Behandelen

In plan Behandelen kan de gebruiker een voorkeur voor een type statine aangeven. Vervolgens wordt deze statines voorgeschreven, middels een user-performed plan. Na drie maanden (dit is gespecificeerd met een time-annotation) wordt plan **Follow-up** geactiveerd, tijdens welke de dosis eventueel verdubbeld wordt. Plan Behandelen eindigt met een user-performed actie waarin jaarlijks de patiënt gecontroleerd wordt.

7.3 Bevindingen tijdens het modelleren in Asbru

Deze paragraaf beschrijft enkele ervaringen die specifiek betrekking hebben op het modelleren in Asbru. Aspecten van Asbru die ook *PROforma* betreffen zullen in hoofdstuk 8 besproken worden.

Gebruik van Asbru is ingewikkeld

Asbru kent veel constructies, daardoor duurde het vrij lang voordat de taal echt begrepen werd. Er zijn bijvoorbeeld veel elementen beschikbaar die de uitvoering en volgorde van uitvoering kunnen beïnvloeden, nl. filter-, setup- en complete-conditions, wait-for, wait-for-optional-children, abort-if, retry-abort-children, on-abort, time-annotation, intention, enz. Daarnaast

is de gebruikte syntax vrij ingewikkeld. Om het protocol nog enigszins begrijpelijk te houden is gekozen om af te stappen van de XML-syntax en een kortere notatie te gebruiken, zonder informatieverlies.

Onduidelijkheid over performance

Het was onduidelijk hoe bepaalde Asbru-constructies te interpreteren. Dit komt door het ontbreken van een interpreter voor de taal: je kunt niet testen of bepaalde constructies doen wat ze moeten doen. De rol van intentions en preferences was hierdoor bijvoorbeeld onduidelijk. In het protocol werden in de intentions van de plannen dan ook alleen de effecten gespecificeerd die duidelijk uit de plannen volgden. De preferences werden in het geheel niet gebruikt.

Onduidelijk onderscheid tussen parameters en variabelen

Bij plan Bepaal-risico zijn `geslacht-positie`, `diabetes-positie`, `smoking-position` en `hypertensie-positie` variabelen (dit is omdat de desbetreffende waarden afhangen van een conditie (if-statement); bij data-abstractie kan echter geen if-then-else gebruikt worden), terwijl `leeftijd-boven-positie`, `leeftijd-onder-positie`, `ratio-boven-positie` en `ratio-onder-positie` parameters zijn (want je wilt zoveel mogelijk detail-informatie verborgen houden, middels data-abstractie). Je hebt dus te maken met gegevens van hetzelfde ‘semantische’ type terwijl ze in verschillende datatypen zijn gedeclareerd.

Data-abstractie

Bij de bepaling of een patiënt hypertensief is wordt kwalitatieve data gebruikt, nl. wel of geen verhoogde bloeddruk en wel of niet behandeld voor hypertensie. Omdat in Asbru data-abstractie op basis van kwalitatieve data niet mogelijk is, moest dit in een plan middels een if-then-else-statement gemodelleerd worden. Deze definitie van hypertensie is echter detail-informatie wat eigenlijk niet in een plan thuishoort, maar op de achtergrond afgeleid dient te worden. Daarnaast kan data-abstractie slecht kwalitatieve data opleveren en dus geen integers. Om omslachtige oplossingen te voorkomen is hier in het protocol echter wel van uitgegaan.

Hoofdstuk 8

Evaluatie

Na analyse van de syntax en semantiek van *PROforma* en Asbru en de in de twee talen gemodelleerde protocollen van de cholesterolrichtlijn kunnen de twee talen vergeleken worden. De verschillen tussen *PROforma* en Asbru zullen in dit hoofdstuk worden beschouwd aan de hand van de uitdrukkingskracht, de begrijpelijkheid van de talen, de mogelijkheden tot hergebruik van onderdelen en het gebruik van beide talen. Het hoofdstuk eindigt met een discussie over de resultaten.

8.1 Uitdrukkingskracht

Bij het modelleren van de cholesterolprotocol in *PROforma* en Asbru en na analyse van de representatie van de kennistypen in beide talen werd duidelijk dat Asbru een grotere uitdrukkingskracht heeft dan *PROforma*. De belangrijkste argumenten hiervoor zullen in het onderstaande worden besproken:

- De procedurele kennis van *PROforma* -protocol wordt gevormd door een soort stroomdiagramrepresentatie. De representatie in de vorm van een graaf is echter onvoldoende wanneer temporele onzekerheid betreffende het begin en eind van taken moet worden uitgedrukt ([7]). Zo is het bijvoorbeeld niet mogelijk om een taak in *PROforma* een bepaalde tijd te laten wachten alvorens het te starten (in een recente versie van *PROforma* is dit wel het geval) en kent *PROforma* geen expliciete constructies om de maximale duur van een taak weer te geven. In de meer declaratieve taal Asbru is dit wel het geval. Asbru gaat hierdoor ook beter om met optionele stappen: een plan waarin een optioneel deelplan is bevat stopt niet perse bij voltooiing van alle verplichte deelplannen, zoals in *PROforma*. In Asbru kan nl. met een *time-annotation* en de optie *wait-for-optional-subplan* worden aangegeven of en hoe lang op voltooiing van de optionele plannen gewacht wordt.
- In een *PROforma*-plan kunnen slechts ‘high-level’ taken voorkomen, terwijl een Asbru-plan naast plannen ook elementaire bewerkingen, zoals variabele-waarde assignments en if-then-else-statements, kunnen bevatten. De vele berekeningen voor het bepalen van het 10-jaarsrisico bijvoorbeeld konden hierdoor in Asbru op een compacte en logische wijze gerepresenteerd worden. In *PROforma* werden deze berekeningen m.b.v. taken gerepresenteerd, en dit resulteerde op dit punt dan ook in een omslachtig protocol-fragment. Hiervan abstraherend lijkt bij de ontwikkeling van *PROforma* de nadruk te hebben gelegen op de representatie van expliciete handelingen en stappen van artsen,

met als gevolg dat de representatie van impliciete, elementaire bewerkingen onderbelicht is gebleven.

- Asbru biedt uitgebreide mogelijkheden tot analyse en abstractie van numerieke parameters. Hierbij is het mogelijk om de verandering van een parameter te berekenen of om uit een numerieke parameter kwalitatieve informatie te abstraheren. Hierbij is het bovendien mogelijk om een context voor de abstractie te specificeren. In Asbru is dit alles gespecificeerd in de definitie van parameters. In *PROforma* bestaan hiervoor geen aparte constructies: dit alles zal gemodelleerd moeten worden m.b.v. bijvoorbeeld decisions, precondities en de default-waarden van data-items. Hierdoor is het moeilijk om dit soort kennis eenduidig en op één plek in het protocol weer te geven.
- Het formaat waarin condities en intentions in Asbru worden gerepresenteerd biedt de mogelijkheid om condities en intentions gestructureerd weer te geven. Een time-annotation kan hierbij worden gebruikt om een temporele context m.b.t. deze expressies uit te drukken. Intentions kunnen bovendien gedetailleerd worden uitgedrukt aan de hand van verschillende dimensies: of de bewering een actie of een toestand betreft, of deze bewering tijdens of na het desbetreffende plan moet gelden en of de bewering bereikt, gehandhaafd of vermeden moet worden. Tenslotte kan in het effect van een plan, waarin het effect van het plan op een bepaalde parameter wordt uitgedrukt, de waarschijnlijkheid en de context voor dit effect worden weergegeven. *PROforma* kent geen gedetailleerd formaat waarmee bovenstaande annotaties kunnen worden onderscheiden van de beweringen waarop deze betrekking hebben; dit alles zal in een enkele expressie gemodelleerd moeten worden. Een uitspraak als 'de temperatuur was de laatste twee uur hoger dan 38 graden' zal bijvoorbeeld gerepresenteerd worden als 'min(temp during [now-hours(2),now]) > 38'. De betekenis van de verschillende elementen in een expressie m.b.t. tot die expressie is hierdoor niet direct duidelijk.
- Verder is de uitdrukingskracht van Asbru ook nog op enkele andere punten uitgebreider dan *PROforma*. Het bleek in *PROforma* bijvoorbeeld onmogelijk om, in tegenstelling tot in Asbru, uit te drukken dat bij het uitvoeren van een groep taken de volgorde niet van belang is. Zo'n groep taken zal in de regel parallel of sequentieel worden gerepresenteerd, waardoor ongewenst de indruk wordt gewekt dat de volgorde wel van belang is. Daarnaast is het in Asbru mogelijk om, geïnitieerd door de suspend-condition, de uitvoering van een plan tijdelijk te onderbreken.

Een sterk punt van *PROforma* vormt de mogelijkheid om een beslissing als een enkele taak, nl. met een *decision*, te representeren. Zo'n beslissing kiest uit enkele kandidaten, waarbij voor elke kandidaat de voor- en tegenargumenten kunnen worden gespecificeerd alsmede het gewicht van deze argumenten. Dergelijke beslissingen zullen in Asbru worden beschreven door gebruik te maken van bijv. if-then-else-statements, variabele-waarde assignments en/of data-abstractie, waardoor het minder duidelijk is op welk punt in het protocol de beslissing wordt genomen.

Het modelleren van tabellen bleek in beide talen problemen op te leveren. Dit is opvallend omdat in richtlijnen vaak tabellen worden gebruikt. In Asbru is het wel mogelijk om lijsten en records te modelleren. Daarnaast viel het op dat beide talen geen expliciete constructies kennen waarmee het gelijktijdig eindigen van stappen gerepresenteerd kan worden.

Tabel 8.1 geeft samenvattend weer hoe *PROforma* en Asbru zich tot elkaar verhouden m.b.t. de kennistypen uit hoofdstuk 2. Bij het vergelijken van de representatie van de kennistypen

in beide talen viel op dat de rol die deze representaties spelen bij de uitvoering van de protocollen in beide talen niet altijd dezelfde is. Het geldig zijn van een goal in *PROforma* bijvoorbeeld heeft tot gevolg dat de desbetreffende *PROforma*-taak wordt overgeslagen. De rol van de vergelijkbare intentions in Asbru is minder direct: deze worden gebruikt bij de selectie van plannen en de evaluatie van de uitvoering daarvan. De reden hiervoor lijkt dat beide talen voor de uitvoering van protocollen niet dezelfde executiemodellen veronderstellen: een *PROforma*-protocol staat op zich, terwijl een Asbru-protocol geactiveerd kan worden vanuit andere protocollen. Bovendien hoeft een arts de aanbevolen stappen in een Asbru-protocol niet per se op te volgen zolang de handelingen van de arts beantwoorden aan de intentie van de aanbevolen stappen.

In het onderstaande zullen voor enkele kennistypen de verschillen in representatie in beide talen toegelicht worden.

Procedurele kennis

Bij *PROforma* blijkt een optionele taak in de praktijk verplicht te zijn wanneer het gevolgd wordt door verplichte taken. Dit is omdat bij sequentieel gemodelleerde taken een taak pas begint wanneer een voorgaande, optionele of verplichte, taak voltooid is.

Preconditie

Het modelleren van een preconditie gebeurt in *PROforma* middels het attribuut *precondition* en in Asbru met een *filtercondition*. In Asbru is het daarnaast ook mogelijk om een preconditie te specificeren die pas na verloop van tijd hoeft te gelden: de *setup-condition*.

Voortijdig beëindigen van stappen

In beide talen kan met een *abort-condition* worden gespecificeerd wanneer een stap af te breken. In *PROforma* wordt een plan met een afgebroken deeltaak zelf niet afgebroken. In Asbru daarentegen kan worden aangegeven welke deelplannen het afbreken van het ouder-plan kunnen veroorzaken.

Het succesvol beëindigen van een stap voordat alle deelstappen voltooid zijn wordt in *PROforma* met een *termination-condition* gerealiseerd. De *complete-condition* in Asbru krijgt pas betekenis wanneer de verplichte deelplannen reeds voltooid zijn, en beïnvloedt dus slechts de uitvoering van optionele plannen.

Postconditie

De postconditie wordt in *PROforma* gerepresenteerd als een variabele-waarde assignment in het attribuut *postcondition*. Dit kunnen ook variabelen zijn die niet gedeclareerd zijn. Aan zo'n variabele kan in het verdere verloop van het protocol gerefereerd worden. In Asbru wordt element *effects* gebruikt om aan te geven welke kwalitatieve invloed de uitvoering van een plan heeft op bepaalde parameters. Net als intentions worden effects gebruikt bij de selectie van een plan.

8.2 Begrijpelijkheid

Voor een taal voor het modelleren van richtlijnen is het belangrijk dat deze begrijpelijk is voor medici. Dit is omdat protocollen vaak door medici beoordeeld worden en uiteindelijk ook door hen zullen worden gebruikt. Een protocol in een voor medici duidelijk gerepresenteerd formaat zal de kennisacquisitie voor de ontwikkeling van het protocol en het gebruik

KENNISTYPEN	PROforma	ASBRU
Doel	d.m.v. <i>goal</i>	d.m.v. <i>intentions</i>
Atomaire stap	<i>action, enquiry</i> en <i>decision</i>	<i>ask</i> en <i>user-performed</i>
Decompositie van stappen	ja	ja
Procedurele kennis		
sequentieel	ja	ja
parallel	ja	ja
synchronisatie	ja	ja
tegelijk eindigen	slechts impliciet	slechts impliciet
cyclisch	ja	ja
conditioneel vertakken	slechts impliciet	ja
optionele stappen	gedeeltelijk	ja
Preconditie	d.m.v. <i>precondition</i>	d.m.v. <i>filtercondition</i> en <i>setup-condition</i>
Voortijdig beeindigen van stappen		
onsuccesvol	d.m.v. <i>abort-condition</i> . Het parent-plan wordt echter niet afgebroken	d.m.v. <i>abort-condition</i>
succesvol	d.m.v. <i>termination-condition</i>	d.m.v. <i>complete-condition</i> en <i>continuation specification</i>
Postconditie	d.m.v. <i>post-condition</i>	d.m.v. <i>effects</i>
Temporele kennis		
absolute en relatieve tijd-referentie	ja	ja
verschillende tijdschalen	ja	ja
temporele eisen stappen	gedeeltelijk	ja
temporele onzekerheid	nee	ja
Representatie van data		
elementaire data-structuren	gedeeltelijk	gedeeltelijk
meerdere waarden per variabele	ja	ja
afleiden van data	d.m.v. een <i>derivation-rule</i> van een data-item en de kandidaten van een <i>decision</i>	d.m.v. data-abstractie en variabele-waarde assignments

Tabel 8.1: Overzicht van de verschillen in kennistypen tussen PROforma en Asbru.

van het protocol bevorderen. Voor ontwikkelaars van protocollen is een duidelijk overzicht van de mogelijkheden van modelleertalen en van het uiteindelijke protocol belangrijk. Hierbij kan een taal die voor het modelleren van een bepaald richtlijnfragment slechts een beperkt aantal alternatieven biedt het mogelijk maken eenduidige en begrijpelijke protocollen te ontwikkelen. Omdat de taaleigenschappen die een taal vanuit het oogpunt van medici begrijpelijk maken overlappen met de taaleigenschappen die de taal vanuit het oogpunt van ontwikkelaars begrijpelijk maken, worden deze eigenschappen in deze paragraaf gezamenlijk besproken. Op dit punt biedt PROforma enkele belangrijke voordelen boven Asbru:

- De procedurele kennis van een PROforma-protocol wordt gevormd door een soort stroomdiagramrepresentatie. Hierdoor kan een protocol in PROforma, in tegenstelling tot in Asbru, worden weergegeven middels een graaf. Een dergelijke representatie is herkenbaar, ook voor niet-informatici, en zal hierdoor bijdragen tot het beter begrijpen van het protocol.
- In Asbru-plannen kunnen constructies voorkomen, zoals variabele-waarde assignments en lijst-manipulaties, die niet direct aansluiten bij paradigmata uit de medische wereld. In PROforma daarentegen zijn alle stappen in voor artsen herkenbare taken, nl. enquiry, action, decision en plan, beschreven; hierdoor wordt er geen beroep gedaan op kennis van reguliere programmeertalen.
- In vergelijking met PROforma beschikt Asbru over veel taal-constructies waarmee procedurele kennis tot in de kleinste details kan worden uitgedrukt, zoals de *propagation specification*, de *time-annotation* en de optie *wait-for-optional-subplans*. Hoewel de uitdrukingskracht in Asbru op dit punt groter is (en dus meestal de juiste alternatieven aanwezig zijn waardoor je niet zelf constructies hoeft te maken), is het op de juiste manier modelleren van de procedurele kennis van een richtlijn tijdrovender dan in PROforma. Bovendien is een dergelijke complexe weergave van procedurele kennis minder goed leesbaar.
- Het modelleren in de Asbru-syntax levert diepe nestingen op, waardoor het resulterende protocol moeilijk leesbaar is. Een PROforma-protocol is minder diep genest.
- Het onderscheid dat in Asbru is gemaakt tussen parameters en variabelen gaf problemen. Voor sommige data was het onduidelijk of deze middels parameters of middels variabelen gemodelleerd moesten worden. Daarnaast kwam het voor dat gegevens van hetzelfde ‘semantische’ type in de verschillende data-typen parameter en variabele waren gedeclareerd, wat verwarrend kan zijn. De redenen hiervoor waren dat variabelen niet kunnen optreden als resultaat van data-abstractie en dat data-abstractie gebaseerd op een niet-numerieke parameter niet mogelijk is. PROforma daarentegen gebruikt slechts één type data-structuur voor het representeren van gegevens, nl. een data-item.

Een belangrijk nadeel van PROforma m.b.t. bovenstaande is dat, in tegenstelling tot Asbru, data-abstractie niet mogelijk is. Hierdoor zullen bijvoorbeeld ook rekenkundige bewerkingen en abstracties van data in PROforma-taken gemodelleerd moeten worden, terwijl je zulke gedetailleerde en voor het begrip van het protocol minder relevante informatie liever op de achtergrond verborgen houdt. Bijkomend gevolg is dat een PROforma-protocol hierdoor relatief (in vergelijking met het aantal plannen in een Asbru-protocol) veel taken bevat.

Daarnaast kent PROforma alleen globale variabelen (data-items). Dit maakt het op dit punt moeilijk om een PROforma-protocol te lezen, omdat hierbij dan vele variabelen onthouden

moeten worden. In Asbru kunnen variabelen lokaal t.o.v. elk plan gedeclareerd worden, waardoor deze buiten zo'n plan niet zichtbaar zijn.

De stroomdiagramrepresentatie van *PROforma* heeft ook een nadeel: het is in een grafische weergave van een protocol niet duidelijk of van een groep parallele taken die volgen op een bepaalde taak slecht enkele of alle taken worden uitgevoerd. In hoofdstuk 6 zijn op dit punt enkele aanbevelingen opgenomen.

Beide cholesterolprotocollen werden aan een specialist ter beoordeling getoond. De grafische representatie van het *PROforma*-protocol bleek voor de expert goed te begrijpen en de *PROforma*-taken enquiry, action, decision en plan waren herkenbaar voor de expert. Het Asbru-protocol daarentegen werd door de expert als onduidelijk ervaren. Een belangrijke reden hiervoor kan zijn dat, in tegenstelling tot het *PROforma*-protocol, het Asbru-protocol niet grafisch kon worden gepresenteerd. Daarnaast was bij de presentatie van het *PROforma*-protocol de procedurele kennis gescheiden van de meer gedetailleerde attributenspecificaties, terwijl in het getoonde Asbru-protocol dit onderscheid minder duidelijk was. Een eenvoudiger en minder gedetailleerder representatie van het Asbru-protocol was hier wellicht meer begrijpelijk geweest.

8.3 Hergebruik

Het vele werk dat gemoeid is met het modelleren van richtlijnen maakt het zinvol te kijken naar mogelijkheden tot hergebruik van (fragmenten van) protocollen in beide talen. Een sterk punt van Asbru vormt hierin dat protocollen zijn georganiseerd in een plan-library. De opbouw van de plan-library in Asbru bevordert de hergebruik van protocollen op de volgende manieren:

- In Asbru wordt ook het gehele protocol als een plan gerepresenteerd. Hierdoor kan een protocol (lees: plan) of een deelplan van dit protocol ook vanuit andere plannen in de plan-library geactiveerd worden. Hierbij is het bovendien mogelijk een plan aan te roepen met een argument (parameter-passing). Ook in *PROforma* wordt een protocol als een plan gerepresenteerd, echter in deze taal staat een protocol op zich. De organisatie van protocollen maakt het onmogelijk om vanuit een protocol naar (delen van) een ander protocol te refereren.
- In Asbru is het mogelijk om variabelen en parameters onafhankelijk van protocollen in de plan-library te definiëren. Hierbij kunnen bovendien variabelen die afhankelijk zijn van een bepaald domein worden onderscheiden van variabelen die voor alle domeinen van toepassing zijn. In *PROforma* zijn de data-items slechts van toepassing binnen het protocol waarin ze gedefinieerd zijn.

Daarnaast kunnen in Asbru verschillende onderdelen van een plan, zoals condities en intentions, gelabeld worden. Hierdoor kan op andere posities in het protocol naar deze onderdelen verwezen worden, waardoor ruimte en ontwikkeltijd wordt bespaard. Dit is niet mogelijk in *PROforma*.

De nadelen van *PROforma* op het gebied van hergebruik vielen tijdens het modelleren van de cholesterolrichtlijn niet zo sterk op als uit bovenstaande verondersteld zou kunnen worden. Wanneer er in beide talen verscheidene richtlijnen gemodelleerd dienen te worden die bovendien complexer zijn dan de cholesterolrichtlijn, zullen de verschillen in de mogelijkheden tot hergebruik wellicht duidelijker worden.

8.4 Gebruik van *PROforma* en Asbru

In deze paragraaf zal worden gekeken naar gebruik en mogelijke toepassingen van beide talen. Enkele, m.b.t. dit aspect relevante, argumenten voor het gebruik van richtlijnen en modelleertalen, zoals genoemd in hoofdstuk 2, zullen als leidraad dienen:

Ondersteuning door de computer bij medisch handelen

Voor *PROforma* is inmiddels software beschikbaar waarmee het protocol ontwikkeld (met Arezzo Composer) en getest (met Arezzo Tester) kan worden. Het modelleren van het protocol met behulp van Composer heeft als voordeel dat meteen duidelijk is of bepaalde taalconstructies geoorloofd zijn. Daarnaast kunnen met Tester protocollen in de praktijk worden getest. Hierbij dient opgemerkt te worden dat in dit onderzoek een beta-versie van de Arezzo-software gebruikt is: er traden geheugenproblemen op bij grote protocollen waardoor met de software niet het gehele protocol gemodelleerd kon worden. Voor Asbru is het nog onduidelijk of en in hoeverre de taal interpreteerbaar is. Hoewel alle elementen een streng formaat kennen is het voor bepaalde elementen uit de taal, zoals intentions, niet mogelijk om de rol die deze bij de uitvoering van een protocol spelen exact te voorspellen.

Toepassing voor medisch onderwijs

De in vergelijking met Asbru betrekkelijke eenvoudige taal, het gebruik van bij medici bekende representaties van taken, de grafische weergave van protocollen en de beschikbaarheid van testsoftware maakt dat in *PROforma* richtlijnen op een begrijpelijke wijze gepresenteerd kunnen worden. Hierdoor zijn deze richtlijnen, meer dan bij gebruik van Asbru, bruikbaar voor bijvoorbeeld medisch onderwijs.

Signaleren van gebreken in richtlijnen

De uitdrukingskracht van Asbru is groter dan van *PROforma*. Hierdoor zullen bij het modelleren in Asbru eventuele onvolledigheden en incorrectheden in de ‘handgeschreven’ richtlijn eerder opvallen.

Bij het modelleren van de cholesterolrichtlijn bleek de richtlijn inderdaad niet geheel correct: de drempelwaarden kwamen niet geheel overeen met de risicotabel. Ook was op sommige punten kennis van experts nodig om de richtlijn aan te vullen. Daarnaast viel het op dat de experts een computerprogramma gebruikten dat afwijkt van de risicotabel en de drempelwaarden.

Evaluatie van de handelingen van een arts

Een protocol in *PROforma* ondersteunt een arts bij het uitvoeren van een richtlijn door automatisch de volgende stap in een richtlijn te selecteren. In Asbru worden daarnaast ook de uitgevoerde handelingen van de arts vergeleken met de intenties van plannen. Hierdoor heeft een arts, zolang de handelingen beantwoorden aan de intentie van het protocol, de vrijheid af te wijken van de voorgeschreven stappen.

8.5 Discussie

PROforma en Asbru lijken beide geschikt voor het modelleren van medische richtlijnen. Aan beide talen ligt een klinisch model van stappen en volgorde van stappen ten grondslag. Procedurele kennis kan worden uitgedrukt, en het is mogelijk om temporele kennis weer te geven. Daarnaast is het formaat van beide talen zodanig dat het, direct of indirect, interpreteerbaar is door een computer. Deze eigenschappen maken de talen in theorie ook geschikt voor

niet-medische doeleinden. Zo zijn voor Asbru inmiddels de mogelijkheden onderzocht voor de toepassing van deze taal bij het modelleren van sporttraining-programma's ([8]), met positieve resultaten.

De uitdrukingskracht vormt een sterk punt bij Asbru, en met name bij het modelleren van complexe richtlijnen. De mogelijkheden tot data-abstractie maken Asbru in het bijzonder geschikt voor richtlijnen waarin veel en vaak numerieke gegevens verwerkt moeten worden, zoals richtlijnen voor intensive-care. Daarnaast biedt het sterk declaratieve karakter van de taal en het element time-annotation de mogelijkheid om richtlijnen te modelleren waarin temporele onzekerheid omtrent de te nemen stappen bestaat. Hierdoor, en door het gebruik van intentions en effects, is de handelingsvrijheid van een arts bij het uitvoeren van een protocol in Asbru groter dan in *PROforma*. *PROforma* kent op het gebied van uitdrukingskracht enkele tekortkomingen. Met name het vertraagd starten van taken werd als een opvallend gebrek ervaren. Daarentegen beschikt *PROforma* met de taak decision over een constructie om beslissingen te nemen op basis van argumenten (zoals productie-regels), waarbij onzekerheid omtrent deze argumenten met gewichten kunnen worden uitgedrukt. Voor het nemen van zulke beslissingen kent Asbru geen gestructureerde constructie.

In Asbru zijn er meer mogelijkheden tot hergebruik van (delen van) protocollen dan in *PROforma*. Dit aspect is onderbelicht in *PROforma*, wat remmend kan werken op de ontwikkeling en verspreiding van protocollen. Een sterk punt van *PROforma* daarentegen is dat deze taal eenvoudiger is en meer aansluit op begrippen uit de medische wereld. De stroomdiagram-representatie van protocollen maakt dat deze makkelijk te begrijpen zijn. Het beschikbaar zijn van software is daarnaast ook een sterk punt van *PROforma*.

Hoofdstuk 9

Conclusie

In de gezondheidszorg worden, ter verbetering van de kwaliteit van medische handelingen, in toenemende mate richtlijnen gehanteerd. Deze richtlijnen schrijven gestructureerd een reeks stappen en de volgorde tussen deze stappen voor, alsmede de condities onder welke de stappen moeten worden uitgevoerd. Naast kwaliteitsverbetering zijn richtlijnen o.a. nuttig bij medisch onderwijs en de evaluatie van de handelingen van artsen.

Het formaat waarin richtlijnen doorgaans worden gerepresenteerd, zoals tekst, tabellen, enz., maakt het moeilijk om richtlijnen om te zetten voor gebruik door beslissings-ondersteunende systemen. Daarnaast zijn richtlijnen die in zo'n ad hoc formaat gerepresenteerd zijn, vaak dubbelzinnig en onvolledig. Het representeren van richtlijnen in zgn. *modelleertalen* of *representatietalen* biedt hierboven voordelen. Deze talen hebben de eigenschap dat de typen kennis die doorgaans in een medische richtlijn voorkomen, zoals medische handelingen en beslissingen, gemodelleerd kunnen worden in een zodanig formeel formaat zodat interpretatie door een computer mogelijk wordt. Een belangrijk aspect daarnaast is dat bij het modelleren van richtlijnen in deze talen eventueel dubbelzinnigheden en onvolledigheden aan het licht komen.

In deze scriptie is een vergelijkend onderzoek tussen de twee modelleertalen *PROforma* en *Asbru* beschreven. Er werd onderzocht hoe verschillende, in medische richtlijnen veel voorkomende, typen kennis worden gerepresenteerd in beide talen. Daarnaast werd een specifieke richtlijn in beide talen gemodelleerd. Hierdoor konden *PROforma* en *Asbru* met elkaar worden vergeleken m.b.t. de uitdrukkingskracht, de begrijpelijkheid, het hergebruik en de toepassing van beide talen. Daarnaast konden voor beide talen enkele aanbevelingen worden gedaan.

Bijlage A

Analyse van de cholesterolrichtlijn

A.1 Toegangscriteria voor de richtlijn

De richtlijn richt zich op personen die een verhoogd risico hebben op het krijgen van een CHZ. De volgende twee groepen kunnen geïdentificeerd worden:

- patiënten met HVZ
- patiënten zonder HVZ maar met van de volgende risicofactoren:
 - CHZ bij eerste graadsfamilieleden jonger dan 60 jaar
 - diabetes mellitus
 - hypertensie, al dan niet behandeld

Bij een patiënt zonder HVZ, diabetes mellitus of CHZ bij eerste graadsfamilieleden jonger dan 60 jaar, waarbij het onbekend is of deze hypertensief is zal de bloeddruk gemeten moeten worden om te bepalen of de patiënt wordt toegelaten.

Op grond van de kosteneffectiviteit van een eventuele behandeling worden binnen deze groep patiënten uitgesloten van behandeling:

- patiënten met een levensverwachting minder dan 5 jaar
- patiënten met manifest hartfalen
- patiënten met een belangrijk verminderde linkerkamerfunctie
- patiënten met een andere ernstige ziekte (naar inzicht van de specialist)
- patiënten van 70 jaar of ouder (mannen) of 75 jaar of ouder (vrouwen)

Patiënten die op grond van bovenstaande punten niet worden uitgesloten ondergaan een lichamelijk onderzoek en een eventuele behandeling (zie A.2 en verder). Bij patiënten met HVZ wordt dit *secundaire preventie* genoemd; bij patiënten zonder HVZ maar met risicofactoren heet dit *primaire preventie*. De patiënten die worden uitgesloten van behandeling vallen verder buiten de richtlijn. Het is vervolgens aan de gebruiker (lees: specialist) wat er met deze patiënten moet gebeuren.

A.2 Lichamelijk onderzoek

Bij primaire en secundaire preventie worden de volgende handelingen verricht:

1. Een bloedonderzoek.
Hierbij worden in ieder geval de volgende waarden gemeten:

- TC-concentratie (TC staat voor Totale Cholesterol)
- HDLc-concentratie (HDLc staat voor 'High-Density'-Lipoproteïnecholesterol)
- Triglyceride-concentratie

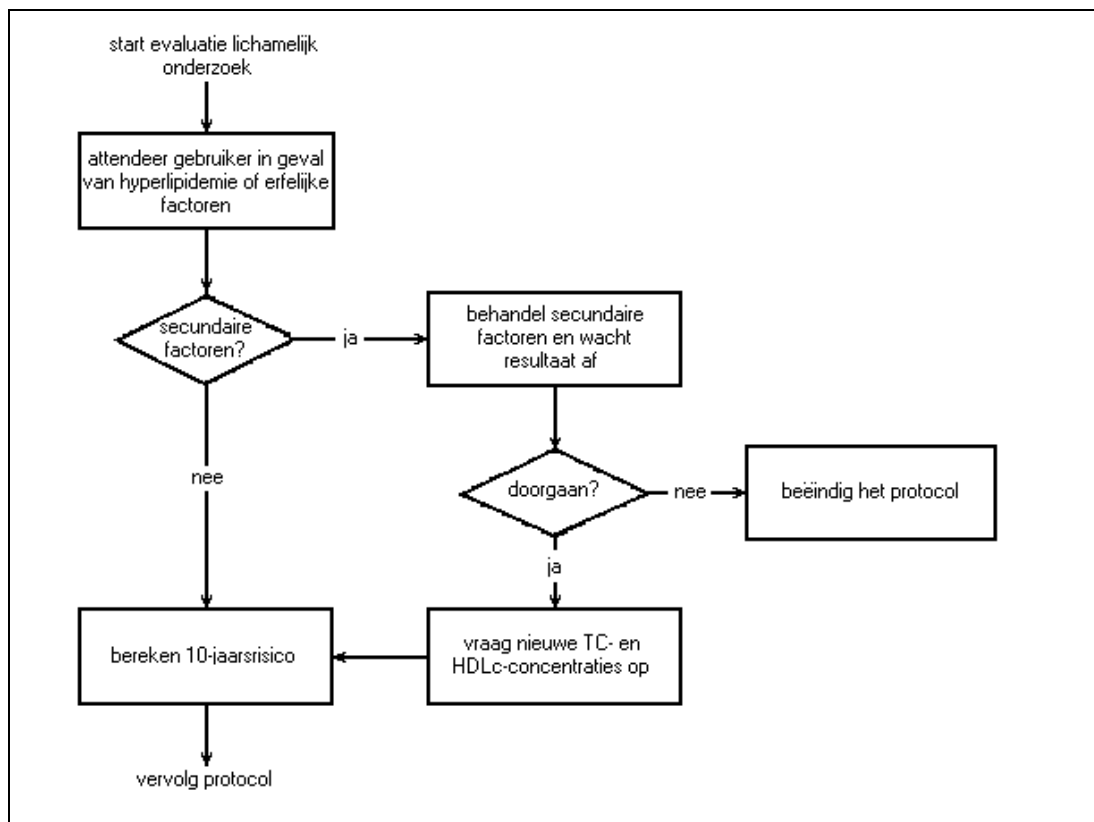
Dit onderzoek betreft een laboratoriumonderzoek en wordt verder niet in de richtlijn uitgewerkt; de gebruiker krijgt de desbetreffende concentraties van het laboratorium. Wanneer tijdens de eerste metingen de TC-concentratie groter is dan 5 wordt door de gebruiker een nieuw laboratoriumonderzoek aangevraagd. De metingen worden vervolgens gemiddeld. Uit de verkregen waarden worden vervolgens de volgende waarden berekend:

- LDLc-concentratie, volgens de *Friedewald*-formule: $\text{LDLc-cholesterol} = \text{TC-conc.} - \text{HDLc-conc.} - (0.45 \times \text{triglyceride-conc})$ (LDLc staat voor 'Low-Density'-Lipoproteïnecholesterol)
 - TC/HDLc-ratio
2. Een (lichamelijk) onderzoek waarbij gezocht wordt naar indicaties op familiale hyperlipidemie.
Bij het onderzoek naar indicaties op familiale hyperlipidemie (d.w.z. erfelijke factoren die een hoog cholesterolgehalte kunnen verklaren) wordt er gelet op *xanthomen*, *xanthelasmata* en *arcus corneae bij patiënten jonger dan 40 jaar*. Dit onderzoek wordt niet in de richtlijn uitgewerkt; er wordt eenvoudig aan de gebruiker gevraagd of er op basis van dit onderzoek indicaties op familiale hyperlipidemie zijn.
 3. Een onderzoek waarbij gezocht wordt naar secundaire oorzaken.
Bij het onderzoek naar secundaire oorzaken (d.w.z. omgevingsfactoren die een hoog cholesterolgehalte kunnen verklaren, zoals overmatig alcoholgebruik of metabole ziekten, zoals diabetes mellitus) wordt er gekeken naar *ALAT*, *gamma-GT*, *TSH*, *glucose* en *eiwit in urine*. Evenals het onderzoek naar indicaties op familiale hyperlipidemie wordt dit onderzoek niet in de richtlijn uitgewerkt; er wordt eenvoudig aan de gebruiker gevraagd of er op basis van dit onderzoek secundaire oorzaken gevonden zijn.

A.3 Resultaten uit het lichamelijk onderzoek

De resultaten uit het lichamelijk onderzoek worden vervolgens geëvalueerd. De volgende gevallen zijn mogelijk:

- Patiënten met secundaire oorzaken worden verwezen naar een specialist (dit kan de gebruiker zelf zijn), met het doel de secundaire oorzaken te behandelen. De gebruiker kan na verloop van tijd aangeven of de richtlijn moet worden afgebroken (wanneer behandeling van de secundaire oorzaken voldoende blijkt) of voortgezet (wanneer behandeling van de secundaire oorzaken onvoldoende blijkt). In het laatste geval zullen opnieuw de TC- en HDLc-concentraties gevraagd worden.
- Bij primaire preventie en een TC-concentratie groter dan 8 mmol/L en/of een triglycerideconcentratie groter dan 4 mmol/L is er sprake van hyperlipidemie. De gebruiker wordt hierop geattendeerd.
- Bij secundaire preventie en een TC-concentratie groter dan 5 mmol/L is er sprake van hyperlipidemie. De gebruiker wordt hierop geattendeerd.
- Bij gevonden erfelijke factoren wordt de gebruiker hierop geattendeerd.
- In de resterende gevallen wordt direct het 10-jaarsrisico bepaald.



Figuur A.1: Stroomdiagram van de evaluatie van het lichamelijk onderzoek.

Figuur A.1 illustreert de evaluatie van het lichamelijk onderzoek. Volgens de CBO-consensus moeten patiënten met gevonden erfelijke factoren, secundaire oorzaken en/of hyperlipidemie verwezen worden naar een specialist. Omdat dit protocol in de regel reeds door specialisten wordt uitgevoerd, wordt de gebruiker slecht geattendeerd op deze bevindingen.

A.4 Bepaling van het 10-jaarsrisico

De beslissing of het 10-jaarsrisico bepaald moet worden is afhankelijk van de resultaten uit het lichamelijk onderzoek, zoals die boven beschreven zijn (na behandeling van secundaire oorzaken wordt de richtlijn nl. eventueel afgebroken). Hoewel bij secundaire preventie het 10-jaarsrisico geen rol speelt bij de beslissing of statines moeten worden voorgeschreven, wordt deze als extra informatie toch berekend. Het 10-jaarsrisico wordt in de CBO-richtlijn bepaald door uit de risicotabel het 10-jaarsrisico af te lezen aan de hand van de leeftijd, het geslacht, wel of niet diabetes, wel of niet roken, wel of niet hypertensief en de TC/HDLc-ratio. Het risico is in de tabel weergegeven voor de leeftijden 40, 50, 60 en 70 jaar, en voor de TC/HDLc-ratiowaarden 4, 5, 6 en 7.

Voor de leeftijden tussen de decaden en tussenliggende waarden voor de TC/HDLc-ratio dient volgens de CBO-richtlijn een schatting te worden gemaakt d.m.v. interpolatie. Het ligt echter meer voor de hand om het risico exact te berekenen i.p.v. te schatten. Op het UMCU wordt inmiddels door specialisten een computerprogramma gebruikt dat het risico berekent. De

risico's die door dit programma worden berekend wijken echter af van de risico's die aan de hand van de risicotabel, m.b.v. interpolatie, worden berekend, en wel om de volgende redenen:

1. Het feit of er CHZ in de familie voorkomt is hier van invloed op de hoogte van het risico.
2. Het feit of patiënt HVZ heeft is hier van invloed op de hoogte van het risico.
3. De hoogte van de systolische bloeddruk beïnvloedt progressief de hoogte van het risico. Hier wordt dus de hoogte van de bloeddruk ingevoerd, i.p.v. wel of geen hypertensie.

Besloten is om op dit punt de CBO-consensus aan te houden, echter met het verschil de tussenliggende waarden te bereken i.p.v. te laten schatten door de gebruiker. De reden voor dit verschil is dat hierdoor het risico nauwkeuriger kan worden bepaald.

A.5 Beslissing wanneer statines voor te schrijven

Na de berekening van het 10-jaarsrisico wordt er besloten of er statines moeten worden voorgeschreven. Een patiënt krijgt statines voorgeschreven wanneer er sprake is van HVZ (dus in het geval van secundaire preventie) of wanneer het 10-jaarsrisico boven een bepaalde drempel uitkomt (in het geval van primaire preventie). Deze drempel is afhankelijk van leeftijd, geslacht, diabetes mellitus en CHZ bij eerste graadsfamilieleden jonger dan 60 jaar. Patiënten met een TC kleiner dan 5 mmol/L of een LDLc kleiner dan 3,2 mmol/L komen niet in aanmerking voor medicamenteuze behandeling.

De CBO-consensus hanteert de volgende drempelwaarden voor het voorschrijven van statines:

- 25% bij 40- en 50-jarigen
- 30% bij 60-jarigen
- 35% bij 70-jarige vrouwen
- 40% bij 70-jarige mannen

Deze drempelwaarden worden in de volgende gevallen verlaagd:

- met 5% bij patiënten met diabetes mellitus
- met 5% bij patiënten met CHZ bij eerste graadsfamilieleden jonger dan 60 jaar
- met 10% bij patiënten met diabetes mellitus en CHZ bij eerstegraadsfamilieleden jonger dan 60 jaar

In de tabel is ook zonder de drempelwaarden af te lezen of voor een patiënt statines moeten worden voorgeschreven. Dit wordt nl. in de tabel weergegeven door afwijkende kleuring van de hoog-risicogeveallen. Toch is hier gekozen om dit expliciet met behulp van de drempelwaarden te bepalen. Hiervoor zijn twee redenen. Ten eerste biedt het de mogelijkheid de gebruiker uit te leggen waarom een patiënt resp. wel of geen statines krijgt voorgeschreven. Ten tweede kan op deze manier ook voor de tussenliggende waarden worden berekend of behandeling met statines geïndiceerd is. De tabel verschilt echter op één punt met de drempelwaarden, nl. bij een rokende man van 70 jaar met diabetes mellitus en zonder hypertensie. Deze heeft een 10-jaarsrisico van 29% en krijgt volgens de tabel cholesterolverlagende medicijnen voorgeschreven ondanks een drempelwaarde van 30%. In dit geval is de indicatie uit de tabel aangehouden.

Het computerprogramma waarmee op het UMCU het 10-jaarsrisico wordt berekend, wordt ook gebruikt om te bepalen of statinebehandeling geïndiceerd is. Echter ook hier verschilt het programma van de CBO-aanpak:

1. De drempelwaarde is hier niet leeftijdsafhankelijk: voor elke leeftijd geldt een drempelwaarde van 25%.
2. In het geval van diabetes mellitus wordt de drempelwaarde niet verlaagd, zoals dit in de CBO-consensus wel het geval is.

Het is onduidelijk waarom het computerprogramma bij de indicatiestelling afwijkt van de CBO-consensus. Op dit punt is in ieder geval indicatiestelling uit de CBO-consensus aangehouden, omdat voor de consensus een breder draagvlak mag worden verondersteld dan voor het computerprogramma.

A.6 Behandeling met statines

De behandelend arts kan bij de keuze van statines naar eigen voorkeur kiezen uit *Simvastatine* en *Prevastatine* (beide 20-40mg). Na drie maanden wordt opnieuw de TC-concentratie gemeten; wanneer deze groter dan 5 mmol/l is wordt de dosis verdubbeld, anders wordt de dosis gecontinueerd. Vervolgens wordt de patiënt vanaf het laatste bezoek jaarlijks gecontroleerd. Deze jaarlijkse controle valt verder buiten deze richtlijn: er wordt eenvoudigweg aan de gebruiker gemeld dat de patiënt jaarlijks gecontroleerd dient te worden.

A.7 Opmerkingen

Hoewel de CBO-richtlijn adviseert om voedingsadviezen en voorlichting over roken te geven, werd dit niet in opgenomen tijdens het modelleren. De reden hiervoor is dat deze adviezen door de experts als triviaal worden beschouwd: voorlichting over roken en voedingsadviezen behoren nl. tot de gebruikelijke activiteiten wanneer een specialist geconfronteerd wordt met patiënten met vaatziekten en/of risicofactoren. Bovendien bestond er onduidelijkheid wanneer en door wie de adviezen worden gegeven.

Bibliografie

- [1] J. FOX, N. JOHNS, A. RAHMANZADEH, 'Protocols for medical procedures and therapies: A provisional description of the PROforma language and tools', in: E. Keravnou, C. Garbay, R. Baud, J. Wyatt (eds.), *Proceedings of the Artificial Intelligence in Medicine, 6th Conference on Artificial Intelligence in Medicine Europe (AIME-97)*, Springer, Berlijn, 1997, pp. 21-38.
- [2] J. FOX, N. JOHNS, A. RAHMANZADEH, 'Disseminating medical knowledge: The PROforma approach', *AI in Medicine*, **14**(1-2), 1998, pp. 157-182.
- [3] A. VOLLBREGT, A. TEN TEJJE, F. VAN HARMELEN, J. VAN DER LEI, M. MOSSEVELD, 'A study of PROforma, a development methodology for clinical procedures', *AI in Medicine*, **17**, 1999, pp. 195-221.
- [4] Y. SHAHAR, S. MIKSCH, P. JOHNSON, 'The Asgaard Project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines', *AI in medicine*, **14**, 1998, pp. 29-51.
- [5] S. MIKSCH, 'Plan management in the medical domain', *AI communications*, **12**(4), 1999, pp. 209-235.
- [6] R. KOSORA, S. MIKSCH, 'Visualization techniques for time-oriented, skeletal plans in medical therapy planning', in: W. Horn, Y. Shahar, G. Lindberg, S. Andeassen, J. Wyatt (eds.), *Proceedings of the Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM'99)*, Springer, Berlijn, 1999, pp. 291-300.
- [7] R. KOSORA, S. MIKSCH, Y. SHAHAR, P. JOHNSON, 'Asbruvew: capturing complex, time-oriented plans - beyond flow-charts', in: *Proceedings of the Second Workshop on Thinking with Diagrams 1998 (TwD '98)*, University of Wales, Aberystwyth, Groot-Brittannië, 1998, pp. 119-126.
- [8] S. MIKSCH, K. HAMMERMÜLLER, 'Asbru, a Plan-Representing Language Modelling Time-oriented, Skeletal Plans in Sport, in: Baca, A. (ed.), *Proceedings of the 2nd International Symposium Computer Science in Sport*, 1999.
- [9] L. OHNO-MACHADO, J.H. GENNARI, S.N. MURPHY, N.L. JAIN, S.W. TU, D.E. OLIVER, E. PATTISON-GORDON, R.A. GREENES, E.H. SHORTLIFFE, G. OCTO BARNETT, 'The Guide-Line Interchange Format: A model for representing guidelines', *Journal of the American Medical Informatics Association*, **5**(4), 1998, pp. 357-372.
- [10] S.W. TU, M.A. MUSEN, 'A flexible approach to guideline modelling', *AMIA Annual Symposium*, Washington, D.C., 1999, pp. 420-424.
- [11] E. PATTISON-GORDON, J.J. CIMINO, G. HRIPCSAK, S.W. TU, J.H. GENNARI, N.L. JAIN, R.A. GREENES, 'Requirements of a shareble guideline representation for computer applications', Stanford University, nr. SMI-96-0628, 1996.
- [12] 'Behandeling en preventie van coronaire hartziekten door verlaging van de plasmacholesterolconcentratie - Consensus cholesterol', Nederlandse Hartstichting, Den Haag, 1998.
- [13] P.C.G. SIMONS, Y. VAN DER GRAAF, A. ALGRA, 'SMART-studie - De prevalentie van 'tweede' vaatziekte bij patiënten met een 'eerste' vaatziekte: heeft screening zin?', Julius Centrum voor Patiëntgebonden Onderzoek, Academisch Ziekenhuis Utrecht, 1998.