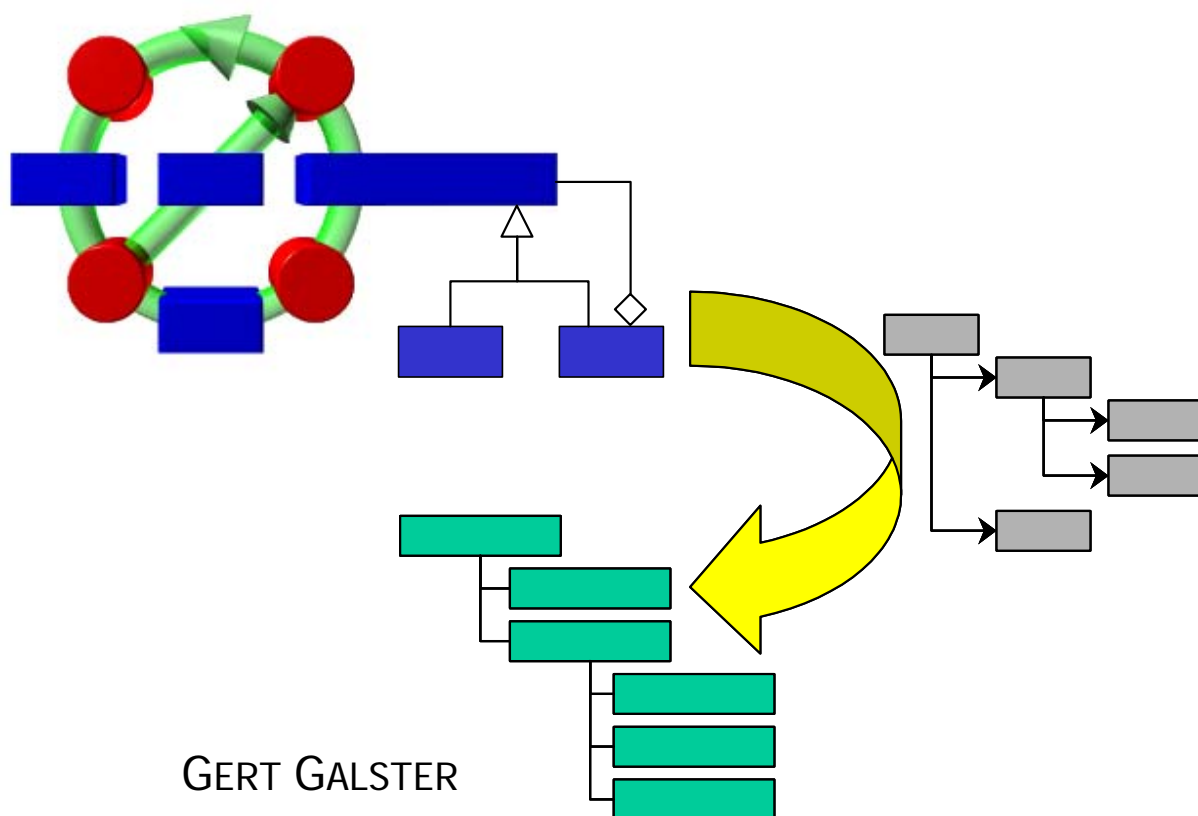


Har arketyper en plads i G-EPJ?



2004

Resumé

Resumé på dansk

I oktober 2002 udgav EpjObservatoriet sin årlige statusrapport, som under overskriften "Modelmæssigt paradigmeskift" indeholdt et substantielt afsnit om anvendelse af arketyper. Det gav straks anledning til diskussion om, hvad arketyper egentlig er, og hvorvidt de har en plads i [GEpj](#). I dette projekt er denne diskussion fortsat. Jeg har gennem litteraturstudier forsøgt at komme til en forståelse af konceptet om anvendelse af arketyper, og jeg har forsøgt at vurdere den bagvedliggende metodes anvendelighed for udviklingen af GEpj.

Jeg konkluderer, at arketypebaseret modellering er et spændende og visionært bud på en udviklingsmetode for sundhedsfaglige IT-systemer, men at det også repræsenterer nogle udfordringer - fortrinsvis med versionering og performance. Endvidere at arketypebaseret modellering kan anvendes i GEpj-udviklingen uden at dette er til ulempe for de, der skal bygge GEpj-baserede IT-systemer.

Det samlede projekt ligger online på adressen:

<http://galster.dk/gert/docs/siproj3/>

Summary in English

October 2002 the Danish EHR Observatory published its annual status report. A substantial section headed 'Paradigm shift in modelling' addresses the application of archetypes. Immediately, this resulted in a discussion about what archetypes really are, and whether they have a role to play in the G-EHR.

This paper continues this discussion. Through studies of relevant literature I have tried to gain understanding regarding the idea of using archetypes, and I have tried to evaluate the applicability of the basic method with respect to development of the G-EHR.

I conclude that archetype based modeling is an exiting and visionary path to pursue regarding a method in developing health care ICT-systems. However, the approach represents a number of challenges, mainly with regard to versioning and performance. Furthermore I conclude that archetype based modelling can be applied in the development of the G-EHR, without it being a disadvantage to those, building a G-EHR based ICT-system.

An online full text version of this paper is available at the address:

<http://galster.dk/gert/docs/siproj3/>

Forord

I nogle år har jeg været ansat i Sundhedsstyrelsens Enhed for Sundhedsinformatik, hvor jeg i væsentlig grad har beskæftiget mig med udviklingen af Grundstruktur for Elektronisk Patientjournal (GEpj).

Dette studieprojekt er ikke formelt udgået fra Sundhedsstyrelsen. Imidlertid er der ingen tvivl om, at det bevidst er lagt i grænselandet mellem studie og arbejde - det skulle være interessant at lave, og det skulle være af betydning for Sundhedsstyrelsens videre strategi for udvikling af GEpj. På denne baggrund er det vigtigt at understrege,

- at meninger og vurderinger, som her kommer til udtryk, ikke kan tages som udtryk for en officiel holdning, men alene er mine egne.
- at der flere steder i arbejdet er trukket på inside-viden - alene fordi der foreligger nogle erfaringer i forbindelse med Gepj-udviklingen, som (endnu) ikke er dokumenteret.

maj 2004
Gert Galster

Indhold

| | |
|--|-----------|
| 1. INDLEDNING..... | 5 |
| 1.1. Baggrund for projektet..... | 5 |
| 1.2. Baggrund for rapporten..... | 6 |
| 1.3. Baggrund for emnet..... | 10 |
| 1.4. Problemformulering | 12 |
| 1.5. Problemafgrænsning..... | 12 |
| 2. TEORI OG METODE | 14 |
| 2.1. Teoretisk referenceramme..... | 14 |
| 2.2. Metoder | 17 |
| 3. HAR "ARKETYPER" EN PLADS? | 21 |
| 3.1. Hvad er "en arketype"? | 22 |
| 3.2. Hvorfor ændre metode? | 22 |
| 3.3. Hvorfor "dual-model"-metoden? | 30 |
| 3.4. Kan arketyper realiseres? | 40 |
| 4. HAR ARKETYPER EN PLADS I GEPJ? | 47 |
| 4.1. Hvad er GEpj? | 47 |
| 4.2. Arketyper i GEpj - nutid eller fremtid? | 52 |
| 4.3. GEpj med arketyper - konsekvenserne | 54 |
| 5. KONKLUSION OG PERSPEKTIVERING | 57 |
| 5.1. Konklusion | 57 |
| 5.2. Vejen fremad... .. | 60 |
| 6. LITTERATUR..... | 63 |

1. Indledning

| | |
|---|-----------|
| 1.1. BAGGRUND FOR PROJEKTET | 5 |
| 1.2. BAGGRUND FOR RAPPORTEN | 6 |
| 1.2.1. Målgruppe | 6 |
| 1.2.2. Ortografi | 6 |
| 1.2.3. Terminologi | 7 |
| 1.2.4. Forkortelser, akronymer & buzz-words | 9 |
| 1.3. BAGGRUND FOR EMNET | 10 |
| 1.4. PROBLEMFORMULERING | 12 |
| 1.5. PROBLEMAFGRÆNSNING | 12 |

1.1. Baggrund for projektet

Dette projekt er lavet som en del af tredje studieår i Sundhedsinformatik under åben uddannelse ved Aalborg Universitet. I projektet er der lagt vægt på at anskue informationsteknologi i sundhedssektoren i et videnskabeligt perspektiv.

Udgangspunktet for projektet har været EpjObservatoriets beskrivelse[1,2] af et modelmæssigt paradigmeskift og af arketyper og i særlig grad observatoriets forslag om, at der "iværksættes et projekt for at undersøge og teste om arketyper kan bruges som harmoniseringsværktøj"[1] i udviklingen af elektronisk patientjournal.

1.2. Baggrund for rapporten

1.2.1. Målgruppe

Målgruppe for denne rapport er sundhedsinformatikere med en væsentlig indsigt i Grundstruktur for Elektronisk Patientjournal ([GEpj](#)).

GEpj er på et overordnet niveau beskrevet i flere tidligere arbejder, hvoraf kan nævnes:

- Galster G. Begrebsmodellen i G-EPJ. April 2002. [Online, HTML]
http://www.sst.dk/Aplikationer/MITOnline/200204/02_Bmodel.htm
! (Verificeret 2004-05-14)
- Galster G. Grundstrukturen – lige nu... Juni 2003. [Online, HTML]
http://www.sst.dk/Aplikationer/MITOnline/200306/02_gepj_aktuelt.html (Verificeret 2004-05-14)

Tilegnelse af rapporten kræver desuden et vist kendskab til Unified Modeling Language (UML), som flere steder - både i denne rapport og i dens kilder - anvendes uden yderligere forklaring.

For information om UML, henvises til

- Fowler M, Scott K. UML Distilled (2nd Ed.). 2000. Addison Wesley Longman
- Object Management Group (OMG) [Online, HTML]
<http://www.omg.org/> (Verificeret 2004-05-14)

1.2.2. Ortografi

Hypertekst: Rapporten er skrevet i hypertekst - dvs med intensiv anvendelse af krydsrefererende hyperlinks. Den **kan** læses på papir-medie, men tilegnelse via elektroniske medier anbefales. Rapporten er elektronisk tilgængelig på adressen <http://galster.dk/gert/docs/siproj3/>

Kildehenvisninger er overalt angivet i kantet parentes, fx. [3] og - så vidt muligt - som hyperlinks. Oversigt over refereret litteratur findes i kapitel 6.

I nogle tilfælde er oplysninger af forskellig årsag ikke tilgængelige for reference. Der er i disse tilfælde anvendt flg ortografi:

- [dokumentation ikke tilgængelig] - oprindelig dokumentation er utilgængelig. Typisk ved overskrivning af Internettets HTML-sider med nyere versioner.
- [upubliceret] - oplysningen fremgår ikke af offentligt tilgængelige kilder, men er baseret på mit personlige kendskab til GEpj-udviklingen
- [personlig meddelelse] - oplysninger som er mig personligt meddelt og som jeg ikke forventer publiceret.
- [personlig observation] - fænomener, jeg selv har iagttaget.

Figurer og grafisk materiale er i billedteksten påført kildehenvisning. Figurer uden kildehenvisning har jeg selv fremstillet.

I nogle tilfælde har kildernes grafiske materiale været af en sådan art eller kvalitet, at genfremstilling har været nødvendig. Der er i disse tilfælde anvendt flg ortografi:

- (efter [99]) - den aktuelle illustration er en genfremstilling af materiale fra reference [99].
- (frit efter [99]) - den aktuelle illustration er en genfremstilling med indholdsmæssige modifikationer af materiale fra reference [99].

1.2.3. Terminologi

I denne rapport er anvendt nogle centrale termer og begreber, som fortjener en eksplicit forklaring:

- Ved **arketype** forstås en model, der udtrykker et domænespecifikt begreb som en begrænsning i instantiering af en underliggende referencemodel. (Oversat fra [8], se afsnit 3.1)
- Ved **begreb** forstås en mental repræsentation af et eller flere virkelige fænomener. (Efter Spri. Metoder och principer i terminologiarbetet, Spri rapport 481. 1999. [Online, PDF] <http://www.spriterm.se/html/rapporter/rap481.pdf> (Verificeret 2004-05-14))

- Ved "**dual-model**"-metode (synonym: arketypebaseret modellering) forstås en metode til systemudvikling, som er baseret på, at systemet beskrives i to samtidige og gensidigt afhængige modeller - [arketype](#) og [referencemodel](#). Konceptet er beskrevet i afsnit 3.3.1.
- Ved **logisk model** forstås en model, som er udtrykt i datalogiske formalismer og som udelukkende afbilder begreber ud fra strukturelle og funktionelle hensyn. For en eksemplificering henvises til afsnittet om logisk model i [28].
- Ved **model** forstås en forsimpning og abstraktion af virkeligheden. (Meget frit efter [Merriam-Webster Online](#). [Online, HTML] <http://www.m-w.com/> (Verificeret 2004-05-14))
- Ved **ontologi** forstås en formaliseret beskrivelse af begreber og relationer mellem disse, som er relevante for et afgrænset emneområde. (Efter Gruber T. [What is an Ontology?](#) [Online, HTML] <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> (Verificeret 2004-05-14))
- Ved **paradigme** forstås en teoretisk referenceramme, indenfor hvilken der er formuleret teorier, love og generaliseringer. (Frit efter [Merriam-Webster Online](#). [Online, HTML] <http://www.m-w.com/> (Verificeret 2004-05-14))
- Ved **referencemodel** forstås en generalisering af logiske modeller med forskelligt sigte¹.
- Ved **term** forstås et symbol for et begreb. Symbolet kan være i form af artikuleret lyd eller bogstaver. (Efter Spri. [Metoder och principer i terminologiarbetet](#), Spri rapport 481. 1999. [Online, PDF] <http://www.spriterm.se/html/rapporter/rap481.pdf> (Verificeret 2004-05-14))

¹ I den refererede litteratur anvendes flere betegnelser for logiske modeller med forskelligt sigte - domæneobjektmodel (DOM), referenceobjektmodel (ROM), referenceinformationsmodel (RIM) mfl. Begrebet referencemodel opfattes som en generalisering af disse.

1.2.4. Forkortelser, akronymer & buzz-words

Det sundhedsinformatiske domæne er oversvømmet af forkortelser, akronymer og buzz-words. Denne rapport er ingen undtagelse:

ADL - uofficiel forkortelse for Archetype Definition Language, se afsnit 3.3.2 og [16,17]

CEN/TC251 - Comité Européen de Normalisation, Technical Committee 251 (CEN/TC251) [Online, HTML] <http://www.cen251.org/> (Verificeret 2004-05-14)

DOM - domæneobjektmodel, se '[referencemodel](#)' i forrige afsnit

EPJ - elektronisk patientjournal

GEpj - Sundhedsstyrelsens Grundstruktur for Elektronisk Patientjournal ([3]), se også afsnit 4.1.

GUI - Graphic User Interface

grundstrukturen - uofficiel betegnelse for [GEpj](#)

NBS - Det Nationale Begrebsråd for Sundhedsvæsnet (NBS) [Online, HTML] http://www.sst.dk/Informatik_og_sundhedsdata/Klassifikationer_og_terminologi/Terminologi/NBS.aspx (Verificeret 2004-05-14)

OpenEHR - The openEHR Foundation (openEHR) [Online, HTML] <http://www.openehr.org/> (Verificeret 2004-05-14)

RIM - referenceinformationsmodel, se '[referencemodel](#)' i forrige afsnit

SKS - Sundhedsvæsenets Klassifikationssystem (SKS) [Online, HTML] http://www.sst.dk/Informatik_og_sundhedsdata/Klassifikationer_og_terminologi/Klassifikationer.aspx (Verificeret 2004-05-14)

SNOMED - Systematized Nomenclature of Medicine (SNOMED) [Online, HTML] <http://www.snomed.org/> (Verificeret 2004-05-14)

UML - Unified Modeling Language, se [tidligere omtale](#).

1.3. Baggrund for emnet

Sundhedsstyrelsen har gennem nogle år udarbejdet en struktureret model for elektronisk patientjournal - GEpj([3]). Sidste udgave udkom i april 2004.

Denne grundstruktur opfattes vidtgående som specifikationen af den nationale standard for elektroniske patientjournaler, som omtalt i den nationale IT-strategi:

[4 pg 35]: Det er aftalt mellem sygehusejerne og regeringen, at de elektroniske patientjournaler bygger på fælles faglige og tekniske standarder, der sikrer, at informationerne mellem systemerne kan udveksles. De elektroniske patientjournaler skal anvende de faglige standarder, som Sundhedsstyrelsen udarbejder, og som afprøves og konkretiseres i samarbejde med amterne/H:S.

I oktober 2002 udgav EpjObservatoriet sin årsrapport for 2002([1]). Heri blev der i flere sammenhænge nævnt et begreb - "arketyper" - som angiveligt skulle være en del af en metode, der tillader decentral udvikling af EPJ. Således skrev EpjObservatoriet:

[1 pg 43]: I perioden indtil alle systemer er baseret på en fælles informationsmodel ... kan man starte med en harmonisering af afgrænsede informationsenheder, det vil sige en harmonisering på arketypleniveau.

Som en del af det team, der udvikler GEpj, læste jeg naturligvis en sådan opfordring med stor interesse. Denne interesse blev yderligere skærpet, da EpjObservatoriet i samme årsrapport anbefalede et projekt for at "undersøge og teste om arketyper kan bruges som harmoniseringsværktøj" i EPJ-modellering[1].

Set i lyset af, at grundstrukturen udgør den nationale standard for EPJ, må EpjObservatoriets anbefaling - som et minimum - kræve, at GEpj kan rumme eller spille sammen med "arketyper". Det har derfor været af interesse at undersøge, hvad "arketyper" er og hvilken plads de kan have i forhold til GEpj.

EpjObservatoriet gav selv nogle bud herpå. I henhold til årsrapporten([1]) og mundtlige indlæg på EpjObservatoriets årskonference([2]) fremstilledes "arketyper" som

- mindre, afgrænsede dataklumper
- små, sundhedsfaglige modeller
- modeller for forskellige velafgrænsede kliniske områder
- en begrebsmodel for et bestemt domæne
- begrebsdefinitioner og-specifikationer for et afgrænset domæne

En sådan fremstilling har næppe en dybde, som gør det muligt at vurdere om "arketyper" kan bruges som harmoniseringsværktøj i national modellering, og det er således skønnet af interesse at belyse, hvad "arketyper" **egentlig** er.

Er de "arketyper", som EpjObservatoriet beskriver, noget, som man blot kan "undersøge og teste"? Netop fordi Observatoriet i sin rapport ser fænomenet i sammenhæng med national harmonisering (læs: GEpj) og fordi Observatoriet eksplicit anfører, at Sundhedsstyrelsens medicinmodul kan være en kandidat til en "arketype"[1], fik jeg en opfattelse af, at "arketyper" allerede findes - eller let kan inkooporeres - i GEpj.

Denne opfattelse var lidt vanskelig at få til at harmonere med, at EpjObservatoriet samtidigt udpeger "arketyper" til at være vejen frem til et modelmæssigt paradigmeskift indenfor den danske EPJ-udvikling[1,2].

I særlig grad blev min opmærksomhed skærpet, da jeg eksplorerede Observatoriets kilder om "arketyper": Den helt centrale forfatter, Thomas Beale, formulerer i "Constraint-based Domain Models for Futureproof Information Systems" ganske klart:

[8]: Executed correctly, a two-level modelling methodology stands to radically change the economics and quality of information systems. As far as is known, none of the existing software development methodologies takes a two-level approach...

og jeg fik således indtryk af, at det omtalte paradigmeskift og anvendelsen af "mindre, afgrænsede dataklumper" er noget, som kan have vidtrækkende både faglige og økonomiske konsekvenser.

1.4. Problemformulering

Set på denne baggrund - med

- en fra EpjObservatoriet udgående anbefaling af et nationalt harmoniseringsprojekt baseret på et løseligt beskrevet fænomen
- hvor anvendelse af det centrale begreb - "arketyper" - må antages at kunne have omfattende faglige og økonomiske konsekvenser
- og hvor det centrale begrebs sammenhæng med den eksisterende nationale standard ikke er overbevisende beskrevet

forekommer det af interesse at beskæftige sig med to centrale spørgsmål:

Hvad er "arketyper"?

Og har de en plads i GEpj?

Denne formulering afspejler ganske nøje den måde, som problemet initialt præsenterede sig - ikke mindst takket være EpjObservatoriets præsentation af "arketyper"[1,2]. Som konsekvens af den proces, som projektet har gennemløbet, står det klart, at formuleringen ikke er præcis: Arketyper er blot en måde at udtrykke en del af produktet fra en bestemt metode til systemudvikling. De essentielle spørgsmål er:

Hvad er arketype-baseret metode?

Og kan GEpj udvikles med denne metode?

1.5. Problemafgrænsning

I dette projekt fokuseres på anvendeligheden af arketypebaseret metode i udviklingen af GEpj. Årsagen til dette fokus er, at EpjObservatoriet udtrykkeligt beskæftiger sig med "arketyper" med henblik på national harmonisering, og GEpj opfattes som den nationale standard. Der er således kun på et ganske overordnet plan taget stilling til, om arketypebaseret metode er anvendelig i andre EPJ-projekter

Der er i dette projekt ikke gjort forsøg på at besvare EpjObservatoriets spørgsmål, hvorvidt "arketyper" kan anvendes som harmoniseringsværktøj i national EPJ-udvikling[1].

Der er i dette projekt ikke taget stilling til andre udviklingsmetoders evne til at kompensere, at IT-systemer skal kunne tilpasse sig et dynamisk problemområde.

I en vurdering af systemudviklingsmetoder for en national standard for elektroniske patientjournaler, trækker konsekvenserne lange skygger. Jeg har valgt at afgrænse "scopet" til udviklingsmiljøet. Spørgsmål af typen:

- Giver denne udviklingsmetode mere brugbare patientjournaler?
- Giver denne udviklingsmetode billigere IT-systemer?
- Er det politisk realistisk at skifte udviklingsstrategi for GEpj?

er således bevidst ignoreret fra dette projekt, jvf afsnit 2.2.3.

At bygge komplekse IT-systemer kræver, at der træffes mange valg på mange niveauer. I dette projekt har jeg valgt at koncentrere mig om highlevel-delen af systemudvikling. Spørgsmål af typen:

- Er Java bedre end C++?
- Hvilken database/middleware skal der udvikles på?
- Kan man lave dynamiske skærmkomponenter?

er således bevidst ignoreret fra dette projekt, jvf afsnit 2.2.3.

2. Teori og metode

| | |
|--|-----------|
| 2.1. TEORETISK REFERENCERAMME | 14 |
| 2.2. METODER | 17 |
| 2.2.1. Valg af referenceramme | 17 |
| 2.2.2. Valg af metode | 17 |
| 2.2.3. Afgrænsning af projekt..... | 17 |
| 2.2.4. Litteratursøgning | 19 |
| 2.2.5. Litteraturanvendelse | 20 |

2.1. Teoretisk referenceramme

Den overordnede teoretiske referenceramme for dette studieprojekt er en model for prospektiv teknologivurdering, som er beskrevet af Christian Nøhr([5]). Med udgangspunkt i en operationel definition af teknologi opstiller Nøhr en metode for prospektiv teknologi-vurdering i otte trin.

Modellen tager sit udgangspunkt i en opfattelse af teknologi som bestående af fire elementer: viden, organisation, teknik og produkt, jvf Figur 1. Et centralt aspekt i denne opfattelse af teknologi er, at de fire elementer er gensidigt afhængige - kvalitative ændringer i ét af elementerne vil give anledning til ændringer i de andre elementer[5].



Figur 1. De basale elementer i teknologi (efter [5])

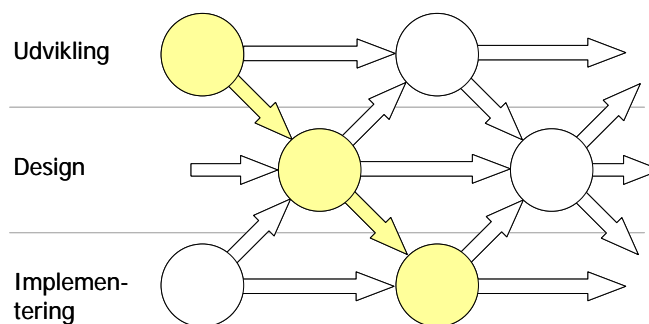
Nøhr beskriver den innovative teknologi-skabende proces som bestående af tre faser[5]:

- **Udviklingsfasen** – hvorunder viden indhentes og teknologiske udviklingskoncepter undfanges. I denne fase skabes centrale dele af teknologiens videns-element.
- **Designfasen** – hvorunder et udviklingskoncept modnes og udmøntes i forskellige tekniske principper og designs, som siden kan anvendes til konstruktion og implementering. I denne fase skabes centrale dele af teknologiens teknik-element.
- **Implementeringsfasen** – hvorunder et design danner grundlag for faktisk konstruktion og anvendelse. I denne fase besluttes teknologiens organisationselement.

I hver af de tre faser foretages der en række valg[5]: Der kan være flere udviklingskoncepter – de er måske ikke alle lige interessante, så der må vælges. Et givent udviklingskoncept kan realiseres gennem forskellige principper og designs, så der må vælges. Et givent design kan både teknisk og organisatorisk realiseres på flere måder, så der må vælges.

Hvert valg indskrænker de frihedsgrader, som er til rådighed i næste fase[5]. Der kan desuden ske en betydelig grad af feed-back[5], som kan påvirke de trufne valg i de enkelte faser.

De beskrevne faser af innovationsprocessen er ikke enkeltstående, men derimod kontinuerte processer[5]. Nøhr beskriver derfor innovationsprocessen som et tre-laget udviklingsforløb (Figur 2):

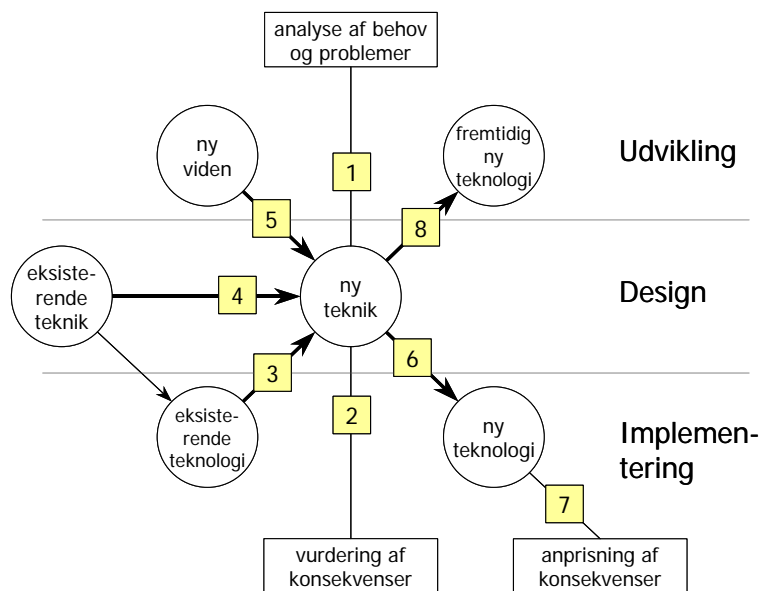


Figur 2. Den innovative proces (efter [5])

På Figur 2 er illustreret innovativ teknologisk udvikling som en kontinuert proces med flere iterationer, hvor hver iteration kan opfattes som et forløb bestående af de oven anførte faser (markeret nedadgående diagonal på Figur 2). Valgene i hver fase er under påvirkning af den forhåndenværende viden om det forudgående forløb.

Nøhr anfører, at prospektiv teknologivurdering bør tage sit afsæt i designfasen, fordi den indeholder det største antal frihedsgrader[5]. Desuden at prospektiv teknologivurdering derved bliver en integreret del af den innovative proces[5].

Med dette udgangspunkt beskrives en metode for prospektiv teknologivurdering i otte trin (Figur 3)[5]:



Figur 3. Trin i prospektiv teknologivurdering (efter [5])

- 1) Analyse af domænets aktuelle behov og problemer og opstilling af specifikke krav til ny teknologi.
- 2) Retrospektiv vurdering af den eksisterende teknologis konsekvenser og målopfyldelse.
- 3) Opsamling af erfaringer med eksisterende teknologi.
- 4) Opsamling af erfaringer med eksisterende teknik.
- 5) Vurdering af om der foreligger relevante udviklingskoncepter.

- 6) Vurdering af hvorledes ny teknik bør konstrueres og implementeres.
- 7) Anprisning af forventede konsekvenser af ny teknologi.
- 8) Angivelse af hvilken ny viden, som bør tilvejebringes for at løse de problemer og behov, som er anført i punkt 1.

2.2. Metoder

2.2.1. Valg af referenceramme

Essentielt drejer dette projekt sig om at afklare fordele og ulemper ved at anvende en ny teknik (["dual-model"-metode](#)) i udvikling af sundhedsfaglige IT-systemer - med særligt fokus på sådanne systemer, som udvikles på basis af GEpj.

Den ny teknik er ... ny - både i international sammenhæng og i forbindelse med udvikling af GEpj-baserede systemer, hvor "dual-model"-metoden endnu ikke har været anvendt. Der er således tale om en prospektiv vurdering af en ny teknik, og det var et oplagt valg at benytte Nøhrs beskrivelse af teknologi og prospektiv teknologi-vurdering ([5], se afsnit 2.1) som teoretisk referenceramme.

2.2.2. Valg af metode

Den af Nøhr beskrevne metode for prospektiv teknologivurdering ([5], se Figur 3) er en udpræget kvalitativ og interaktiv metode, som er skønnet velegnet til at belyse de store, brede spørgsmål, som udgør problemformuleringen i dette projekt. Med udgangspunkt i denne metode er der igennem projektet stillet mindre og mere afgrænsede spørgsmål, som er søgt besvaret gennem litteratur-studier.

2.2.3. Afgrænsning af projekt

At anvende den beskrevne metode for prospektiv teknologi-vurdering som grundlag for dette projekt har dog givet anledning til en række overvejelser om teknologisk "scope": Nøhrs arbejde er baseret på en operationel

definition af teknologi som bestående af fire gensidigt afhængige elementer: viden, organisation, teknik og produkt. Det ligger implicit i denne opfattelse af teknologi, at disse elementer har en fælles afgrænsning - den anskuede teknologi.

I dette projekt, har jeg ønsket at vurdere anvendeligheden af en ny teknik indenfor sundhedsfaglige IT-systemer i bred almindelighed ("Har arketyper" en plads?") og desuden indenfor udviklingen af GEpj ("Har arketyper en plads i GEpj?"). Det har derfor været relevant at anskue teknologi med to forskellige afgrænsninger:

En teknologi vedrører udvikling af GEpj:

| | |
|---------------------|---|
| Viden | Visioner og overordnede mål for sundhedsfaglige IT-systemer. Domænekendskab og indsigt i domænets problemfelter, herunder fx resultater af arbejdsgangsanalyser. Resultater af tidligere teoretiske iterationer (tidligere modeller). Resultater af tidligere praktiske iterationer (prototypeudvikling og pilotforsøg). |
| Organisation | Sundhedsstyrelsens team for GEpj-udvikling. Leverandører af sundhedsfaglige IT-systemer. |
| Teknik | Værktøjer og metoder til overordnet system-beskrivelse (fx klassediagrammer og usecases i UML). |
| Produkt | GEpj |

En anden teknologi vedrører udvikling af sundhedsfaglige IT-systemer:

| | |
|---------------------|--|
| Viden | Samme som ovenfor. Kan omfatte viden om GEpj. |
| Organisation | Overordnet udviklingsteam (fx af openEHR, HL7, GEpj). Leverandører af sundhedsfaglige IT-systemer. Brugere. Borgere/patienter. Sygehusejere. Sundhedsvæsnet. Politikere. (ignoreret, se afsnit 1.5) |
| Teknik | Værktøjer og metoder til overordnet system-beskrivelse (fx klassediagrammer og usecases i UML). Værktøjer og metoder til design og implementering af IT-systemer. (ignoreret, se afsnit 1.5) |
| Produkt | Sundhedsfaglige IT-systemer. |

Denne sondring - baseret på den teoretiske referenceramme - er brugt til at afgrænse projektets omfang, som det fremgår af tabellerne og afsnit 1.5.

2.2.4. Litteratursøgning

Det har været kendetegnende for dette projekt, at langt den overvejende litteratur om emnet findes online uden at være indekseret i akademiske søgebaser.

Fx er der med PubMed-søgning (<http://www.pubmed.gov/>) opnået flg resultater:

| Type | Søgestreng | Total hits | Relevante hits |
|------------|---|------------|----------------|
| MESH | "Medical Informatics/methods"[MeSH] AND "Medical Records Systems, Computerized"[MeSH] | 135 | 0 |
| Ord & MESH | archetype AND "Medical Informatics"[MeSH] | 6 | 0 |
| Ord | "EHR" AND "model" | 21 | 1 |
| Forfatter | Beale T.[Author] | 9 | 1 |
| Forfatter | Kalra D.[Author] | 46 | 3 |

Medens der fx med Google-søgning (<http://www.google.com/>) er opnået flg resultater:

| Type | Søgestreng | Total hits | Relevante hits |
|------|---------------|------------|----------------|
| Ord | archetype EHR | > 680 | > 100 |

Der kan tænkes flere årsager hertil:

- Arketypebaseret modellering er en relativt ny metode, og produktionstiden for hardcopy-publikation og MEDLINE-indeksering kan være årsag til at emnet ikke findes.
- De udvikler-teams, som har beskrevet denne metode lægger måske større vægt på at udvikle EPJ end at lave videnskabelig produktion.
- De udvikler-teams, som beskriver denne metode, gør det måske på et så ringe niveau, at det ikke har fundet plads i videnskabelig publikation.

Det vil overstige projektets ambitionsniveau at gennemføre denne diskussion her. Efter at have vurderet kvaliteten af relevant online-litteratur (og dette projekt er i sig selv en kvalitativ vurdering heraf) og på baggrund af simple kvantitative betragtninger har jeg måttet konstatere, at det i dette projekt har været mere relevant at anvende Internet-baseret søgning fremfor akademiske søgebaser.

Fremfindning af relevant litteratur er derfor helt overvejende sket ad to kanaler:

- Gennem "referencers referencer" - dvs en konsekvent fremfindning af originalt kildemateriale.
- Ved eksploration af en række udvikler-teams' Internet-sites for yderligere relevant materiale, jvf næste afsnit.

2.2.5. Litteraturanvendelse

Under litteraturgennemgangen var det allerede på et tidligt tidspunkt bemærkelsesværdigt, at mængden af lovprisning var stor, hvorimod det var umuligt at finde seriøst kritisk litteratur og ret vanskeligt at finde dokumentation for konceptets holdbarhed i praktisk implementering.

Dette skal ses i sammenhæng med, at alt fremfundet litteratur om arketype-baseret modellering helt åbenlyst er baseret på arbejde udgående fra meget få teams med intensiv krydsbemanding:

| Team \ Forfatter | Beale T. | Kalra D. | Heard S. | Lloyd D. |
|---------------------------------|----------|----------|----------|----------|
| open EHR | X | X | X | X |
| Deep Thought Informatics | X | | | |
| GEHR | X | X | X | X |
| Ocean Informatics | X | | X | |
| CHIME | | X | | X |

På grund af den omtalte skævhed i fundet litteratur har det ikke været muligt at sammenholde litteratur, der afspejlede divergerende opfattelser af arketype-baseret modellering. Jeg har derfor valgt i dette projekt at føre en dialog med den foreliggende litteratur - en kritisk gennemlæsning, hvor jeg har forsøgt at vurdere logikkens stringens og forudsætningernes holdbarhed.

3. Har "arketyper" en plads?

| | |
|---|-----------|
| 3.1. HVAD ER "EN ARKETYPE"? | 22 |
| 3.2. HVORFOR ÆNDRE METODE? | 22 |
| 3.2.1. Argumenterne | 23 |
| 3.2.2. Diskussion | 26 |
| 3.2.3. Konklusion..... | 29 |
| 3.3. HVORFOR "DUAL-MODEL"-METODEN? | 30 |
| 3.3.1. Argumenterne | 30 |
| 3.3.2. Diskussion | 34 |
| 3.3.3. Konklusion..... | 40 |
| 3.4. KAN ARKETYPER REALISERES? | 40 |
| 3.4.1. Hvordan realiserer man "dual-model"-metoden? | 41 |
| 3.4.2. Hvordan er "dual-model"-systemers performance?..... | 44 |
| 3.4.3. Kan "dual-model"-udvikling ske med uafhængige teams? | 45 |
| 3.4.4. Hvordan laver man trefoldig versionsstyring?..... | 45 |

Konceptet om anvendelse af "arketyper" i sundhedsinformatik tager sit udgangspunkt i en kritisk holdning til udvikling af sundhedsfaglige IT-systemer. Med dette udgangspunkt er - fortrinsvis af Thomas Beale - beskrevet en udviklingsmetode, som anvender "arketyper", og som angiveligt skulle være bedre.

I dette kapitel gøres konceptet om anvendelse af "arketyper" til genstand for diskussion:

- **Hvorfor ændre metode?** Hvad er begrundelsen for at skifte udviklingsmetode? Går det ikke meget godt, som det går?
- **Hvorfor "dual-model"-metoden?** Hvad går metoden ud på? Er det teoretisk muligt at bygge systemer, som er konforme overfor to samtidige modeller?
- **Kan arketyper realiseres?** Teori er godt, men fungerer konceptet også i praksis? Hvilke ulemper og faldgruber er der?

3.1. Hvad er "en arketype"?

Ordet "arketype" betyder urform, mønster (Nudansk Ordbog) eller - fra Merriam-Webster Online (<http://www.m-w.com/>):

the original pattern or model of which all things of the same type are representations or copies

Konceptet om anvendelse af arketyper i sundhedsinformatisk software er fremsat af Thomas Beale i slutningen af år 2000[dokumentation ikke tilgængelig]. I den nutidige version 2.2.1 af Beales officielle præsentation af arketyper[6] som en del af openEHR-projektet beskrives begrebet som

[6 pg 28]: ... a constellation of valid combinations of ROM objects for a particular domain concept.

Bedre - og mere dækkende - beskrivelser af begrebet findes i den officielle openEHR-dokumentation

[7]: Archetypes each define (and effectively constrain) legal combinations of the building-block classes defined in the Reference Model for particular clinical domains or organisations by specifying particular record entry names, data-types and prescribed value ranges, and values for some of the context attributes.

og i Beales fremstilling til OOPSLA 2002 workshop[8], som indeholder en eksPLICIT definition:

[8 pg 10]: ... a model defining some domain concept, expressed using constraints on instance structures of an underlying reference model.

der her oversættes til:

Ved **arketype** forstås en model, der udtrykker et domænespecifikt begreb som en begrænsning i instantiering af en underliggende referencemodel.

3.2. Hvorfor ændre metode?

Konceptet om anvendelse af arketyper i sundhedsinformatik tager sit udgangspunkt i en kritisk fremstilling af de fremherskende metoder for udvikling af sundhedsfaglige IT-systemer.

I dette afsnit gøres konceptets forudsætninger til genstand for diskussion: Er der overhovedet et problem?

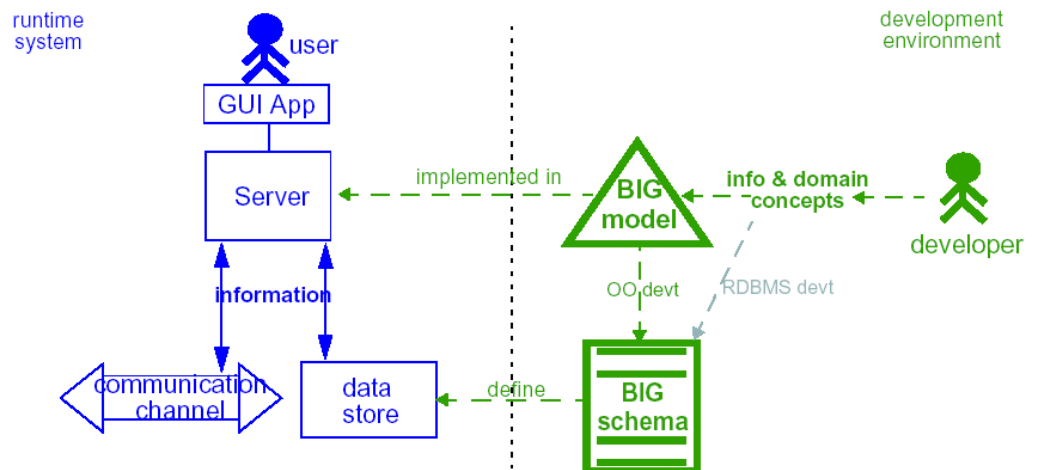
3.2.1. Argumenterne

I henhold til Beales opfattelse af problemfeltet er det sundhedsfaglige domæne ikke bare stort, men ubegrænset ("open-ended")[8] og underkastet betydelig dynamik og variabilitet[6].

I henhold til Beales opfattelse af sundhedsinformatisk software er den overvejende majoritet af applikationer udviklet ved hjælp af klassisk metodologi bestående af iterativ modellering af det samlede problemfelt og i sidste ende omsætning af modellen til konkret applikation[6]. Med Beales egne ord:

[6 pg 11] ... describe system building as an iterative process of writing use cases, finding classes, and building a model which will eventually become software.

Resultatet af denne metode beskriver Beale som et system, hvor mange begreber på meget forskelligt abstraktionsniveau er presset sammen i én stor model - "single-model" (Figur 4) - hvor domænespecifikke begreber hård-kodes ind i applikationerne[6].



Figur 4: "single-model" (fra [6 fig 1])

Den ovenstående Figur 4 illustrerer "single-model"-metoden som to adskilte compartements:

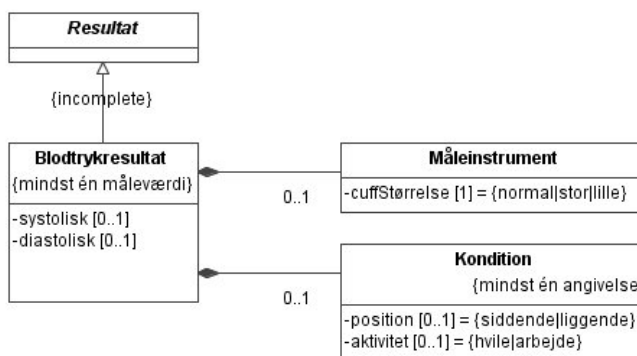
- Til venstre et bruger-compartement (runtime system), hvor brugeren gennem en GUI-grænseflade har adgang til en EPJ-server. Denne server er en implementering af én samlet model, hvor både generiske og fagspecifikke begreber er modelleret side om side. Serveren kan lagre og udveksle data i et format specificeret af den samlede model.
- Til højre et udvikler-compartement (development environment), hvor udviklere (både dataloger og sundhedsfaglige specialister) sammen udvikler den totale model, som applikation og udvekslingsformat baseres på.

"Single-model"-metoden er yderligere illustreret i Eksempel 1:

Eksempel 1. Blodtryk udtrykt med "single-model"

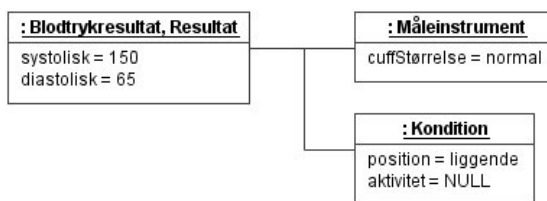
Dette (forsimplede!) eksempel tager sit udgangspunkt i ønsket om at afbilde begrebet "blodtryk" ved hjælp af en referencemodel - helt analogt med den måde, som fx GEpjs medicinmodul er udviklet - jvf Figur 18.

I referencemodellen afbildes det generiske begreb resultat (opfattet som udkomme af aktivitet) som vist på Figur 5:



Figur 5. Referencemodel for blodtryk

Referencemodellen udtrykker, at Blodtryksresultat er en (af mange) specialisering af Resultat. I Blodtryksresultat og associerede klasser er udtrykt de begreber, som var anset af relevans på modelleringstidspunktet.



Figur 6. Et instantieret blodtryksresultat

På Figur 6 er vist et eksempel på en instantiation af blodtryksresultat i henhold til referencemodellen i Figur 5.

(Slut eksempel 1)

Beale ser en svaghed i, at de komplekse domænespecifikke begreber modelleres på samme abstraktionsniveau (jvf Eksempel 1) og med samme værktøjer og kompetencer, som de generiske begreber[6]. Han betvivler desuden, at de formalismer, som er tilgængelige for klassisk modellering er af en sådan kvalitet, at de er egnede til at udtrykke de komplicerede sammenhænge mellem sundhedsfaglige begreber[6].

Det er endvidere Beales opfattelse, at når først en model er implementeret - når først objekt-struktur og databaser ligger fast - så kræver det betydelige ressourcer at udvide eller ændre den underliggende model[6].

Beale anfører desuden en række svag- og uhensigtsmæssigheder[6]:

- Den tilgrundliggende model kan i sagens natur kun afspejle den på udviklingstidspunktet eksisterende indsigt plus de bedste gæt om fremtiden (se Eksempel 1 og prøv at udtrykke "middelblodtryk"). Enhver ændring af modellen - og dermed applikationen - vil kræve betydelige ressourcer, og modellen vil i praksis fremstå som statisk.
- At bygge én samlet model, som både rummer generiske og meget (sundheds-)specifikke begreber stiller krav til udvikler-teamet: dataloger har vanskeligt ved at modellere uden et ekstraordinært kendskab til sundhedsinformatik og sundhedsfaglige eksperter har vanskeligt ved at bidrage til modelleringen uden en ekstraordinær indsigt i datalogisk metode.
- At modellen er statisk og alligevel skal danne underlag for fungerende applikationer rejser et krav om, at modellen på

implementeringstidspunktet skal være udtømmende. Dette forekommer umuligt i et univers, der er så dynamisk som det sundhedsfaglige domæne. Applikationer vil derfor tendere mod at blive obsolete alene fordi deres underliggende model i praksis ikke lader sig opdatere.

- Netop kravet om at modellen er udtømmende på implementeringstidspunktet rejser et uløseligt skizma mellem på den ene side at blive færdig med applikationen og på den anden side få udviklet modellen "i bund". Konsekvensen vil i praksis være enten forsinkede software-projekter eller allerede fra starten insufficiante applikationer. I værste fald både-og.
- En konsekvens af at modellen er statisk er, at modeller bliver proprietære sammen med de berørte applikationer. Konvergens af modeller mellem flere udvekslingscentre vil være besværliggjort.
- At modeller er proprietære vil i betydelig grad vanskeliggøre udveksling af data mellem forskellige applikationer.

Beales løsning på disse problemer er at anvende arketyper - modeller, der udtrykker domænespecifikke begreber som strukturelle begrænsninger af en underliggende referencemodel. Den dertil hørende metode er beskrevet i afsnit 3.3.

3.2.2. Diskussion

Beales argumentation synes at hvile på følgende forudsætninger:

- Det sundhedsfaglige domæne er ubegrænset stort og underkastet betydelig dynamik og variabilitet.
- Den overvejende majoritet af sundhedsinformatisk software er udviklet på en måde, så domænespecifikke begreber hård-kodes ind i applikationen ("single-model").
- Når først en model er implementeret, kræver det betydelige ressourcer at udvide eller ændre den underliggende model.

I dette afsnit gøres disse forudsætninger til genstand for diskussion med henblik på at vurdere, om Beales kritik af sundhedsfaglig IT-udvikling er berettiget.

At det sundhedsfaglige domæne er meget stort er vanskeligt at drage i tvivl. Med godt 350.000 [SNOMED](#)-termer[6] og ca 80.000 danske [SKS](#)-termer[10] er der ingen tvivl om at domænet er ... velvoksnet.

Jeg kan imidlertid overveje, om domænets begreber har den dynamik og variabilitet, som Beale påstår. Denne problematik har været omtalt i et tidligere arbejde[10], som - også med støtte i Rectors fremstilling[11] - beskrev domænet som underkastet polydimensionel ekspansion, fordi

- mange begreber er (endnu) ikke defineret - endsige klassificeret
- domænet har tradition for et meget "løst" sprog med omfattende polysemi og stor vægt på kontekstuel kommunikation
- domænet udvides jævnlige - med nye aspekter eller nye metodikker
- domænet bliver i tiltagende grad fin-granuleret.

Er det korrekt, når Beale hævder at majoriteten af sundhedsinformatisk software har domænespecifikke begreber hård-kodet ind i applikationerne? Det er vanskeligt at forholde sig til et så generaliseret postulat, men i den aktuelle sammenhæng - arketyper i GEpj - forekommer det at være en farbar vej at nøjes med at se på faktisk implementering af GEpj, og stille spørgsmålene:

- Udtrykker GEpj i sin logiske model domænespecifikke begreber?
- Bliver domænespecifikke begreber fra GEpjs logiske model hårdkodet ind i applikationerne?

Det første spørgsmål er besvaret i afsnit 4.1.3, hvoraf det fremgår, at der i GEpj's logiske model er et betydeligt indhold af domænespecifikke begreber.

Det er straks sværere med det andet spørgsmål. I de to seneste projekter til afprøvning af GEpj:

- GEPKA-projektet (<http://medinfo.dk/epj/proj/gepka/>)
- FLPR-projektets AH-del (http://medinfo.dk/epj/proj/flpr_ah/)

var der i begge tilfælde eksplicit forudsat, at applikationerne kunne gengive GEpj's logiske model. Dette er fx i projektbeskrivelsen for GEPKA formuleret:

[12 pg 14]: Evalueringsmålene for prototypen er ... at prototypen udgør en implementering af G-EPJ, herunder af ... RIM'en og forretningslogikken.

Det fremgår desuden, at applikationerne skal kunne udveksle data i henhold til en given XML-syntax[12] (som er baseret på den logiske model).

Men om dette faktisk betyder, at applikationerne hårdkoder (læs: fysisk implementerer) de domænespecifikke begreber, er ingenlunde givet. Erfaringer fra disse projekter antyder, at dette kan være afhængigt af den til rådighed stående softwareplatform[upubliceret], og man kan ikke afvise at i hvert fald en del domænespecifikke begreber implementeres på fysisk niveau.

Et yderligere indicium for, at domænespecifikke begreber faktisk hårdkodes ind i applikationer, kommer fra EpjObservatoriets vurdering af Århus Amts EPJ (dvs ikke GEpj-baseret). I rapporten påpeges, at softwareplatformen som udgangspunkt er meget generisk (faktisk for generisk), men at denne generiskhed vidtgående ikke er anvendt, men tværtimod er koblet ud ved hjælp af hårdkodning[22]. Dette benægtes dog kategorisk af projektledelsen for Århus Amts EPJ[23].

En sidste forudsætning for Beales argumentation for anvendelse af arketyper er, at når først en model udviklet efter "single-model"-metoden er implementeret, så kræver det betydelige ressourcer at udvide eller ændre den underliggende model. Dette er et velkendt fænomen i softwareudvikling, hvor man af samme årsag lægger betydelig vægt på så tidligt som muligt at få foretaget en tilbundsående og fyldestgørende analyse af problemfeltet. Som anført af Biering-Sørensen et al:

[13]: det kan koste op til 100 gange mere at rette en "fejl" i drifts- og vedligeholdelsesfasen frem for i kravspecifikationsfasen.

Anskuet fra udviklersiden (udvikling af GEpj, national udvikling af klassifikationer og klinisk terminologi) bemærkes det, at der er et betydeligt pres for at undgå ændringer - angiveligt fordi ændringer vil koste omfattende ressourcer i form af reprogrammering[upubliceret]. Dette kan eksemplificeres med den vægt, som fx projektledelsen for Århus Amts EPJ tillægger stabilitet og kompleksitet af GEpj:

[23 pg 6]: I udviklingen af GEPKA har Sundhedsstyrelsen indtil videre kommet med 9 opdateringer og her i april 2004 bliver disse samlet i en G-EPJ version 2.0. Yderligere har Sundhedsstyrelsen annonceret endnu en udgave senere i 2004. Århus Amt har helt klart ønsket at afvente udvikling, afestning og endelig klinisk accept af GEPKA inden man indarbejdede en så omfangsrig ændring af den allerede udviklede Integrationsplatform.

3.2.3. Konklusion

Sammenfattende finder jeg, at Beales argumentation for, at den nuværende strategi for udvikling af sundhedsfaglige IT-systemer er problematisk, hviler på troværdige forudsætninger. Selve argumentationen - gengivet i af afsnit 3.2.1 - forekommer på dette grundlag både stringent og logisk, og jeg ser ingen grund til at betvivle, at Beale har ret, når han konkluderer, at modellering af det sundhedsfaglige domæne ved hjælp af metoden "single-model" er uhensigtsmæssig[6].

Jeg har hermed ikke forholdt mig til, om anvendelsen af "single-model"-metoden har den altoverskyggende udbredelse, som Beale beskriver.

3.3. Hvorfor "dual-model"-metoden?

Den ovenfor beskrevne opfattelse af det sundhedsfaglige domæne og de fremherskende metoder for sundhedsfaglig softwareudvikling er af Thomas Beale anvendt som argument for at indføre en ændret udviklings- og applikationsmetode, som benævnes "dual-model"-metoden.

Man kunne forestille sig andre metoder til at kompensere, at IT-systemer skal kunne tilpasse sig et dynamisk problemområde. Som anført i afsnit 1.5 er dette anset at ligge uden for dette projekts rammer.

I dette afsnit gøres beskrivelsen af "dual-model"-metoden til genstand for diskussion: Hvad går konceptet ud på? Er det teoretisk muligt at bygge systemer, som er konforme overfor to samtidige modeller?

3.3.1. Argumenterne

Beale anfører, at problemfeltets begreber kan udtrykkes i to adskilte modeller på hver sit abstraktionsniveau - et generisk og et domænespecifikt[8]. Udkommet af denne adskillelse er således to modeller, hvoraf den generiske model (referencemodellen) kun afbilder begreber, som er stabile og generiske

[8 pg 6]: ... must ... contain only non-volatile concepts in order to be maintainable.

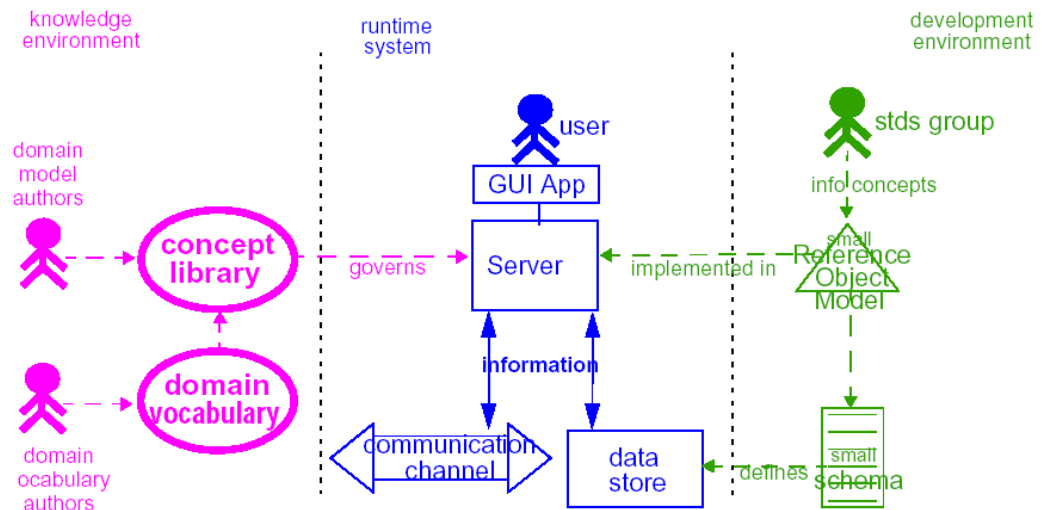
- hvorimod den domænespecifikke model (arketypen) udtrykker problemfeltets variabilitet og dynamik.

[8 pg 6]: ... and is where the numerous, volatile concepts of most domains are expressed.

Denne dobbelte modellering af begreber benævnes af Beale "dual-model"[6], og det er karakteristisk, at referencemodellen - gennem strukturel begrænsning udtrykt i arketypemodellen - danner grundlaget for at udtrykke de domænespecifikke begreber.

Beale anfører desuden, at arketypen kan udtrykkes som en maskin-læsbar ontologi[6].

"Dual-model"-metoden er er illustreret på Figur 7:



Figur 7: "dual-model" (fra [6 fig 6])

Den ovenstående Figur 7 illustrerer "dual-model"-metoden som tre adskilte compartements:

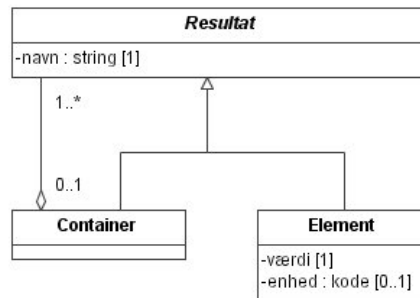
- Til højre et "datalogisk" udviklings-compartement (development environment), hvor udviklere på baggrund af generiske - og dermed stabile - begreber har lavet en simpel model, som fastlægger datastruktur for lagring og informationsudveksling.
- Til venstre et "sundhedsfagligt" udviklings-compartement (knowledge environment), hvor udviklere har udtrykt sundhedsfaglige termer og begreber som maskinlæsbare ontologier.
- I midten et bruger-compartement (runtime system), hvor brugeren gennem en GUI-grænseflade har adgang til en EPJ-server. Denne server tilgår og behandler data, som er struktureret på basis af den simple datalogiske model, men som samtidigt er udtrykt med begrænsninger hentet fra de sundhedsfaglige maskinlæsbare ontologier.

Konceptet om at instantiering tilvejebringes i et samspil mellem to modeller er illustreret i nedenstående Eksempel 2. Eksemplet er konstrueret med henblik på direkte sammenligning med det forrige Eksempel 1.

Eksempel 2. Blodtryk udtrykt med "dual-model"

Dette (forsimplede!) eksempel tager sit udgangspunkt i ønsket om at afbilde begrebet "blodtryk" ved hjælp af en referencemodel og en arketype.

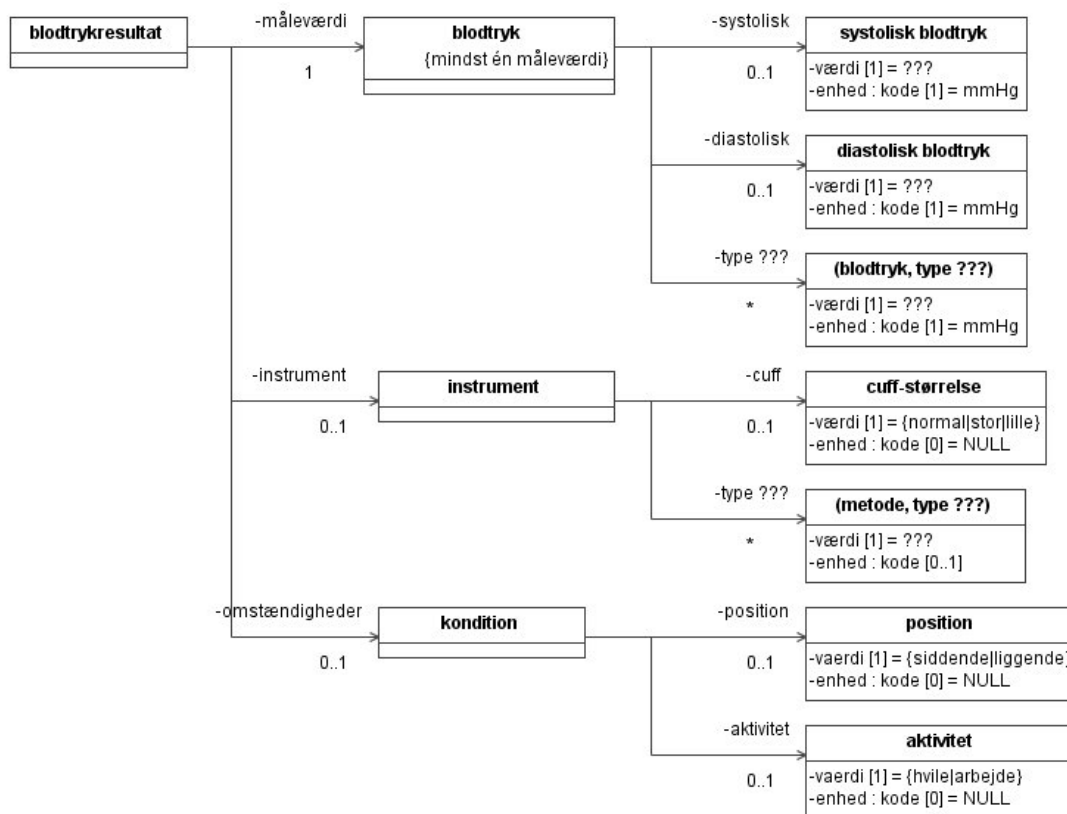
I referencemodellen afbildes det generiske begreb "Resultat" (opfattet som udkomme af aktivitet) som vist på Figur 8:



Figur 8. Referencemodel for resultat

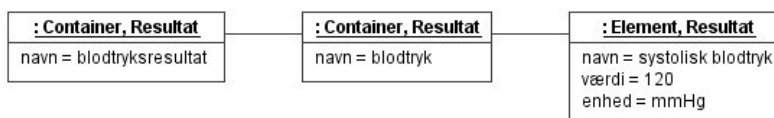
Referencemodellen udtrykker, at Resultat kan indeholde Resultat kan indeholde Resultat - et uendeligt dybt træ af Resultater. Desuden at Resultater, som ikke indeholder yderligere Resultater, har en værdi og muligvis en enhed.

I arketypen afbildes det domænespecifikke begreb "blodtryksresultat" (opfattet som en domænespecifik og variabel instantiering af Resultat) som vist på Figur 9:



Figur 9. Arketype for blodtryk

Arketypen udtrykker de begreber, som var anset af relevans på modelleringstidspunktet, og indeholder desuden pladsholdere for ekstra begreber (blodtryk-type og metode-type).



Figur 10. Et instantieret blodtryksresultat

På Figur 10 er vist et eksempel på en intantiering af blodtryksresultat i henhold til referencemodellen i Figur 8 og arketypen i Figur 9.

(Slut eksempel 2)

Fordelen ved denne tilgang er ifølge Beale[6], at

- en referencemodell, som alene afbilder generiske begreber, er mindre og mere stabil.

- ved effektivt at adskille data-laget (referencemodellen) og videns-laget (arketyper) bortfalder kravet til applikationer om, at de skal være fyldestgørende på udviklingstidspunktet.
- ved effektivt at adskille data-laget og videns-laget sikres, at applikationer er fremtidssikre - det er muligt for en eksisterende applikation at behandle nye/opdaterede arketyper uden at dette kræver massive ændringer.
- udvikling af det samlede system kan varetages af interessenterne på en mere effektiv måde og uden direkte sammenhæng - dataloger vil have en væsentlig indflydelse på udviklingen af referencemodellen, men en lille plads i udviklingen af de domænespecifikke ontologier. Det omvendte vil gælde for domæne-eksperterne.

3.3.2. Diskussion

Konceptet "dual-model" baseres på følgende forudsætninger[6,7,8]:

- Man kan skille problemfeltets begreber i to: de generiske og de domænespecifikke, hvoraf de generiske er stabile i tid, medens de domænespecifikke udtrykker problemfeltets variabilitet og dynamik.
- Man kan af de generiske begreber udtrykke en referencemodel, som gennem strukturel begrænsning kan danne grundlag for at udtrykke domænespecifikke begreber.
- Man kan på struktureret og maskinlæsbar vis udtrykke strukturel begrænsning af en referenceinformationsmodel.

Disse forudsætninger vil i det følgende blive gjort til genstand for diskussion ud fra en rent teoretisk referenceramme.

Kan man indenfor det sundhedsfaglige domæne skille problemfeltets begreber i to: de generiske og de domænespecifikke, hvoraf de generiske er stabile i tid, medens de domænespecifikke udtrykker problemfeltets variabilitet og dynamik?

Som udgangspunkt er adskillelsen af begreber i et binært udfaldsrum (generisk versus domænespecifik) en urimelig forsimpning. Det er let at

forestille sig en kæde af abstraktion, fx: overlæge (er en) læge (er en) sundhedsperson (er en) sundhedsprofessionel (er en) person (er et) levende væsen (er et) begreb. Og hvor er det så lige at skillelinjen mellem "generisk" og "domænespecifik" skal sættes? Det ser ud som om, afgrænsningen af begreber (og dermed domæner) let bliver en arbitrær øvelse. Og helt identiske forhold gør sig gældende i en binær sontring mellem "stabil" og "flygtig".

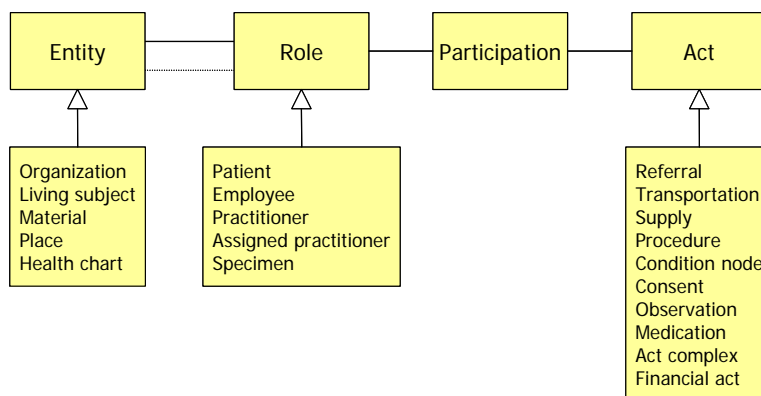
Ved nærlæsning af kilderne [6,7,8] står det klart, at teamet bag openEHR ikke er "fast i kødet" omkring det generiske. Der tales om "generiske" og "ikke-flygtige" begreber uden klar afgrænsning, og jeg efterlades med indtryk af, at der er tale om en principielt arbitrær - men nok pragmatisk ganske velbegrundet - sontring. I openEHR's beskrivelse af principper for design[9 kap 6] er der nævnt en række ganske pragmatiske retningslinjer for netop denne sontring.

Det bemærkes, at netop denne sontring mellem det stabile og det flygtige er essentiel for argumentationen om, at "dual-model"-systemer er lettere at vedligeholde.

Kan man på grundlag af generiske (stabile) begreber lave en referencemodel, som kan danne grundlag for afbildning af domænespecifikke (flygtige) begreber?

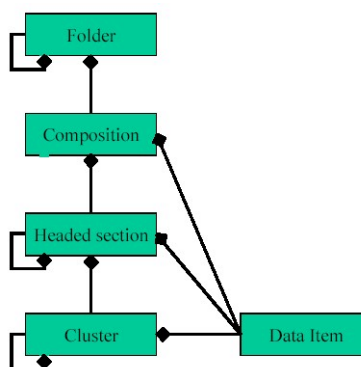
Dette spørgsmål lader sig ikke besvare entydigt indenfor dette projekts rammer, men ved en blot overfladisk undersøgelse af forskellige udviklingsteams overordnede modellering står det klart, at netop denne øvelse er forsøgt af flere, og at de har valgt ganske forskellige tilgange (metaforer) i den overordnede modellering. Dette synspunkt illustreres nedenfor på grundlag af arbejderne fra HL7, CEN/TC251 og GEpj:

HL7 synes at have som udgangspunkt, at sundhedsfaglig dokumentation kan udtrykkes som dokumentation af en række af transaktioner. Den overordnede model ([14], se Figur 11) synes således at udtrykke transaktionen: Nogen gør noget ved nogen ved hjælp af noget [1 pg 61].



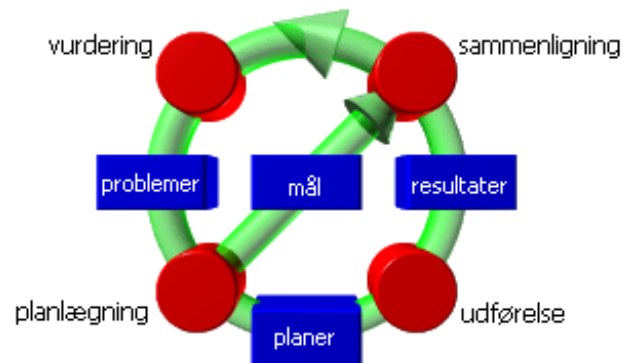
Figur 11. Transaktionstilgangen (HL7) (frit efter [1 pg 61])

CEN/TC251 synes at have som udgangspunkt, at sundhedsfaglig dokumentation kan udtrykkes som en række hierarkisk ordnede dokumenter. Den overordnede model ([15], se Figur 12) synes således at udtrykke, at dokument kan indeholde dokument[1 pg 48].



Figur 12. Dokumenttilgangen (CEN/TC251) (fra [1 pg 48])

GEpj har som udgangspunkt, at sundhedsfaglig dokumentation kan udtrykkes som dokumentation af en række processer, som tilsammen udgør delene i en generisk problemløsende metode. Den overordnede model ([27], se Figur 13) udtrykker således fire generiske processer.



Figur 13. Procestilgangen (GEpj) (fra [27 pg 5])

Den beskrevne diversitet må tages som et væsentligt argument for, at det faktisk er muligt at lave referencemodeller baseret på generiske (stabile) begreber.

Man kan diskutere om sådanne modeller er udtømmende - dvs om de på et abstrakt plan afbilder ethvert begreb af interesse. Dette anses ikke at være en diskussion om modellering, men om valget af den bagvedliggende metafor - fx er HL7-metaforen bedre end GEpj-metaforen? En sådan diskussion anses at gå uden for dette projekts rammer.

Om disse referencemodeller kan danne grundlag for afbildning af domænespecifikke (flygtige) begreber er et andet spørgsmål. I henhold til "single-model"-metoden skulle dette ske i form af specialisering - til et givent domænespecifikt begreb findes der et modelleret overbegreb, som kan specialiseres. I henhold til "dual-model"-metoden skulle dette ske i form af strukturel begrænsning - et givent domænespecifikt begreb kan udtrykkes som en begrænsning af det tilsvarende modellerede overbegreb.

Denne problematik anses at være identisk med den ovenfor omtalte: Er den bagvedliggende metafor sufficient?

Kan man på struktureret og maskinlæsbar vis udtrykke strukturel begrænsning af en referencemodel?

Dette spørgsmål har tydeligvis beskæftiget Thomas Beale og resten af udvikler-teamet bag openEHR en del. I den initiale beskrivelse af arketyper (2000) skriver Beale:

[6]: Here, we will take the structural approach for basic semantics, and use syntactic representation for more complex semantics. The primary justification for the structural approach is that fundamentally we want the AM to relate formally to the ROM, and it is easier to engineer this with two related models than with a model and a language.

To år senere - i 2002 - var der stadig ikke et maskinlæsbart sprog, men man tilsigtede at udvikle det[8], og i slutningen af 2002 fremlagde Beale et al[16,17] et udkast til et maskinlæsbart arketypesprog, om hvilket de skriver:

[16]: ... the possibility of creating an initial archetype language shared by all. ... intended to require as little as possible commitment to specific underlying models ... It is the job of stakeholders to adjust these as required, through discussion and experience.

- dvs et sprog, som stiller visse krav om konformitet til den underliggende referencemodel. Dette sprog - herefter benævnt Archetype Definition Language (ADL) - er bevidst udformet med henblik på både at tilgås maskinelt og humant[17]. Der er på Figur 14 vist et udsnit af en blodtryksarketype (eksempel fra [17]):

```

archetype racgp.openehr.entry.blood_pressue.v1
concept [at0000]
ENTRY[at0000] ∈ {
  name ∈ {CODED_TERM ∈ {definition ∈ {COORDINATED_TERM ∈ {ac0001}}}} -- blood pressure measurement
  data existence ∈ {1} ∈ {
    LIST_S[at1000] occurrences ∈ {1..1} ∈ {
      items existence ∈ {1} ∈ {
        cardinality ∈ {2..*} ordered
        ELEMENT[at1100] occurrences ∈ {1..1} ∈ {
          name ∈ {CODED_TERM ∈ {definition ∈ {COORDINATED_TERM ∈ {ac0002}}}} -- systolic BP
          value ∈ {QUANTITY ∈ {magnitude ∈ {0-1000}; units ∈ {mm[Hg]}}}
        }
        ELEMENT[at1200] occurrences ∈ {1..1} ∈ {
          name ∈ {CODED_TERM ∈ {definition ∈ {COORDINATED_TERM ∈ {ac0003}}}} -- diastolic BP
          value ∈ {QUANTITY ∈ {magnitude ∈ {0-1000}; units ∈ {mm[Hg]}}}
        }
        ELEMENT[at1300] occurrences ∈ {0..*} ∈ {
          name ∈ {*} -- anything
          value ∈ {}
        }
      }
    }
  }
  protocol existence ∈ {0..1} ∈ {
    ...
  }
}

```

Figur 14. En arketype beskrevet i ADL (udsnit fra [17 pg 25])

Som anført af EpjObservatoriet og openEHR har teamet bag EPJ-udvikling i Århus Amt anvendt "dual-model"-metoden[1,18] - se også afsnit 3.4. I den forbindelse har man anvendt en syntax for Hændelsesbeskrivelsesdefinition (HBD), som er illustreret på Figur 15:

| HBD: BtPulsTp Blodtryk Puls Temperatur | | |
|--|---------------------------------|--|
| AND ELEMENTKLYNGE: Blodtryk (0..1) | | |
| 0.0 | Heltal (0..1) Systolisk (mmHg) | |
| 0.1 | Heltal (0..1) Diastolisk (mmHg) | |
| OR ELEMENTKLYNGE: Yderligere detaljer | | |
| 0 | AND ELEMENTKLYNGE: Nej | |
| 0.2 | AND ELEMENTKLYNGE: Ja | |
| 0.2.1 | 0.2.1.0 | Klassificering (0..1) Målested |
| | 0.2.1.1 | Klassificering (0..1) Metode benyttet ved måling |
| | 0.2.1.2 | Streng (0..1) Bemærkninger |
| AND ELEMENTKLYNGE: Puls (0..1) | | |
| 1.0 | Heltal (0..1) Slag pr. minut | |
| OR ELEMENTKLYNGE: Detaljer | | |
| 1 | AND ELEMENTKLYNGE: Nej | |
| 1.1 | AND ELEMENTKLYNGE: Ja | |
| 1.1.1 | 1.1.1.0 | Klassificering (0..1) Målested |
| | 1.1.1.1 | Klassificering (0..1) Målemetode |
| | 1.1.1.2 | Streng (0..1) Bemærkninger |
| AND ELEMENTKLYNGE: Temperatur (0..1) | | |
| 2.0 | Decimaltal Temperatur(°C) | |
| OR ELEMENTKLYNGE: Detaljer | | |
| 2 | AND ELEMENTKLYNGE: Nej | |
| 2.1 | AND ELEMENTKLYNGE: Ja | |
| 2.1.1 | 2.1.1.0 | Klassificering (0..1) Målested |
| | 2.1.1.1 | Streng (0..1) Beskrivelse |

Figur 15. HBD fra Århus-DOM (fra [1 pg 76])

Med disse to eksempler synes det godt gjort, at man - i det mindste på et teoretisk plan - faktisk på struktureret og maskinlæsbar vis kan udtrykke strukturel begrænsning af en referencemodel.

3.3.3. Konklusion

Sammenfattende finder jeg, at argumentationen for at anvende "dual-model"-metoden til udvikling af sundhedsfaglige IT-systemer er baseret på lidt svage teoretiske forudsætninger, idet jeg bemærker at forudsætningen om, at begreber klart kan adskilles i generiske (i referencemodellen) og domænespecifikke (i arketyper), ikke er opfyldt. Man kan derfor ikke på et teoretisk grundlag garantere, at et "dual-model"-system er lettere at vedligeholde end et "single-model"-system. Jeg bemærker dog, at sondringen mellem generiske (stabile) og domænespecifikke (flygtige) begreber kan have en ret stabil pragmatisk løsning, og jeg må på dette grundlag forvente, at et "dual-model"-system faktisk **kan** være lettere at vedligeholde.

Metoden synes - ud fra rent teoretiske overvejelser - at have et betydeligt potentiale.

3.4. Kan arketyper realiseres?

I den dokumentation, som udgår fra teamet bag openEHR (her kilderne [6,7,8,9,16,17,18]) beskrives en række fordele ved "dual-model"-metoden og anvendelse af arketyper. Efter at have forstået konceptet og dets teoretiske rationale efterlades jeg med en række spørgsmål, som nedenstående gøres til genstand for diskussion:

- Hvordan realiserer man "dual-model"-metoden? Foreligger der "proof-of-concept"?
- Hvordan er performance for systemer udviklet med "dual-model"-metoden?

- Præcis **hvor** uafhængige er de teams, som laver arketype og referencemodel?
- Et "dual-model"-koncept rummer to interagerende modeller, som desuden skal "spille sammen med" et klassifikationssystem. Hvordan laver man trefoldig versionsstyring?

3.4.1. Hvordan realiserer man "dual-model"-metoden?

Hvordan laver man et faktisk system, som er konformt overfor to samtidige modeller?

Det er i denne forbindelse fundet relevant at fremhæve to projekter:

- ERDIP-projektet (bidraget fra South & West Devon)
- Projekterne lavet på Århus Amts DOM

ERDIP-projektet (bidraget fra South & West Devon)

I dette delprojekt [19 pg 12f] blev der bygget en "demonstrator" baseret på OpenEHR-modellen - således med anvendelse af "dual-model"-metoden[7] - med det formål at skaffe "proof-of-concept". Denne demonstrator blev fyldt med identitets-slørede data fra 20.000 reelle patienter. Fra konklusionerne bemærkes:

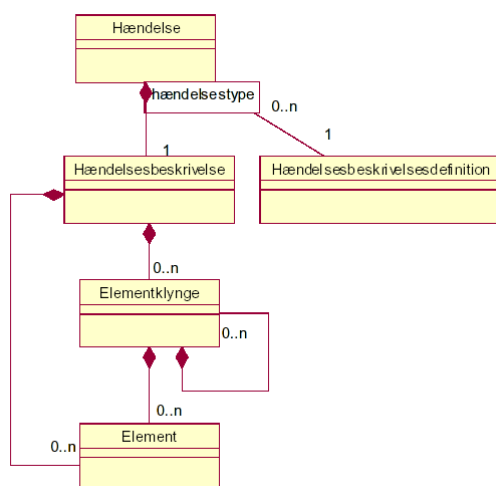
[20 pg 22]: The OpenEHR reference model in conjunction with the use of archetypes for update and query make it possible to implement a health record that has excellent characteristics of integrity and accountability.

og der kan med denne demonstration næppe rejses tvivl om, hvorvidt "dual-model"-metoden er realiserbar.

I samme projekt fremhæves vanskeligheder med at tilvejebringe datakvalitet fra og match med datafødende systemer og med at opnå en tilfredsstillende performance[20]. Problemerne med at få datafødende systemer til at matche og levere kvalitetsdata anses at være temporære (omend med en lang tidsakse)[20]. Problemerne med performance er nærmere omtalt i det følgende afsnit 3.4.2.

Projekterne lavet på Århus Amts DOM

Teamet bag EPJ-udvikling i Århus Amt har - som anført af EpjObservatoriet og openEHR - anvendt "dual-model"-metoden[1,18]. Et af flere steder, hvor dette kommer til udtryk i Århus-DOM'en er i hændelsesdelen af "Procesmodellen" (se Figur 16), hvor indholdet af Hændelsesbeskrivelse, Elementklynge og Element fortolkes ud fra indholdet af Hændelsesbeskrivelsesdefinitionen (HBD)[21].



Figur 16. Detalje fra Århus-DOM (fra [21 pg 15])

Der er på den tidligere Figur 15 vist et eksempel på indholdet af en HBD.

Som anført af EpjObservatoriet præsenterede Århus Amt i oktober 2002 et proof-of-concept hvor en næsten komplet "minimumsjournal" bygget på "dual-model"-metoden blev fremvist, og hvor de dynamiske modelmuligheder blev demonstreret[1].

EpjObservatoriet fandt i 2002 anledning til at omtale Århus-DOM som "arketypelignende" og udtalte eksplicit, at

[1 pg 60]: Århus HBD'er ... er begrebsmæssigt samme størrelser som GEHR's arketyper på forskellige abstraktionsniveauer.

[1 pg 75]: ... Det kan diskuteres, hvorvidt HBD'erne har samme spændvidde i abstraktion som Arketypemodellerne har.

EpjObservatoriet går ikke i detaljer med denne forskel i abstraktionsniveau mellem HBD'er og arketyper, men jeg må antage, at der sigtes til, at hvor arketyper i dokumentationen fra teamet bag OpenEHR beskriver

domænespecifikke begrænsninger af **forskellige** generiske begreber[9], så er i Århus-DOM alle HBD'er en beskrivelse af **det samme** generiske begreb: en hændelse[21].

Denne opfattelse understøttes yderligere af, at EpjObservatoriet i en vurdering af Århus Amts EPJ i april 2004 skriver:

[22 pg 10]: Det anbefales, at integrationsplatformens underliggende generiske model gøres mere sundhedsspecifik ...

Ovenfor er nævnt to projekter, som har realiseret applikationer baseret på "dual-model"-metoden. Det kan - ikke mindst af hensyn til overvejelserne om performance i næste afsnit - være af interesse med nogle mere generelle betragtninger om, hvordan et "dual-model"-system fungerer.

EpjObservatoriet påpeger, at

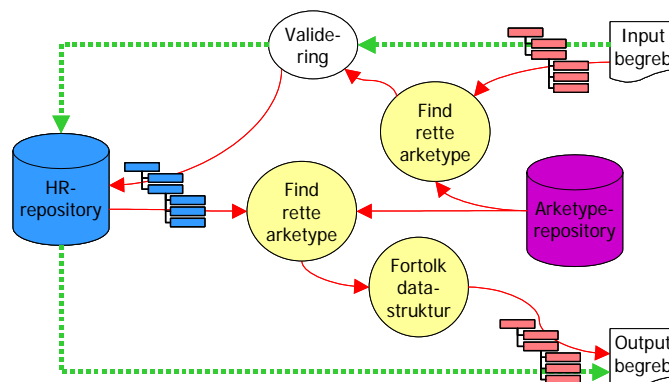
[1 pg 46]: En væsentlig pointe med brug af arketyper er, at de benyttes under kørslen af EPJ-systemet. Det vil sige, at de informationer, brugeren ser på skærmen, er en kombination af de "rå" data og de restriktioner/filtre, der ligger i arketyper.

Det vil sige: der sker en runtime-fortolkning af "rå" data, som de findes lagret i henhold til referencemodellen. Dette er tidligere illustreret på Figur 7 og er den direkte begrundelse for, at arketyper skal kunne udtrykkes maskinlæsbart. I det tidligere nævnte ERDIP-projekt er beskrevet, hvordan arketyper anvendes i runtime-fortolkning[24]. Den nedenstående Figur 17 søger at illustrere dette og se det i forhold til "single-model"-metoden:

På Figur 17 er illustreret en række operationer, som i et "dual-model"-system (vist med tynde, røde pile) nødvendigvis må foregå i realtid:

Ved indlæsning af et begreb - fx blodtryk, jvf Eksempel 2 - skal systemet finde og fortolke den rette arketype. På dette grundlag kan der foretages en validering af data. Hvis data er valide, kan de lagres i henhold til referencemodel og arketype.

Ved udlæsning af et begreb - fx blodtryk, jvf Eksempel 2 - skal systemet finde og fortolke den rette arketype, hvorefter det er i stand til at fortolke meningen med de lagrede data.



Figur 17. Runtime-operationer

På Figur 17 er desuden til sammenligning illustreret de tilsvarende operationer, som i et "single-model"-system (vist med tykke, grønne pile) nødvendigvis må foregå i realtid:

Ved indlæsning af et begreb - fx blodtryk, jvf Eksempel 1 - foreligger data i en kendt struktur og med kendt semantisk betydning. Data skal blot valideres før de lagres i henhold til referencemodellen.

Ved udlæsning af et begreb - fx blodtryk, jvf Eksempel 1 - foreligger data straks i en kendt struktur og med kendt semantisk betydning.

Sammenfattende kan det udtrykkes således: "Dual-model"-metoden stiller krav til systemet om et ekstra overhead til runtime-operationer.

3.4.2. Hvordan er "dual-model"-systemers performance?

Det ovenstående krav om nødvendigt overhead til runtime-operationer træder i relief, når man ser det i sammenhæng med evalueringen af ERDIP-projektet (omtalt i afsnit 3.4.1):

[20 pg 14]: The OpenEHR implementation used by the project ... did not deliver the performance needed for operational use. Although the OpenEHR reference model and the use of archetypes were found eminently fit for purpose, the software implementation of OpenEHR will need to be re-engineered in order for it to be fit for operational use.

Ganske lignende overvejelser var relevante, da Århus Amt i oktober 2002 præsenterede sit proof-of-concept (omtalt i afsnit 3.4.1). Der blev ved denne lejlighed demonstreret responstider, som ville være uacceptable i et

driftssystem[personlig observation]. Endnu i april 2004 skriver EpjObservatoriet om Århus Amts EPJ:

[22 pg 10]: Den største risiko for projektet vurderes fortsat at være integrationsplatformens evne til at levere en tilstrækkelig performance.

I ERDIP-projektet konkluderede man dog, at man ikke så grund til at betvivle at et re-programmeret system ville kunne levere sufficient performance[20].

3.4.3. Kan "dual-model"-udvikling ske med uafhængige teams?

Det er i beskrivelsen af "dual-model"-metoden anført, at den tillader, at systemudvikling varetages et team til referencemodellering og et andet team til arketypermodellering[6]. Fordelen skulle være, at man bedre kan forvalte kompetencerne i de to teams. Men er en sådan opsplitning realistisk?

Det har ikke været muligt at finde litteratur, som belyser dette spørgsmål. Erfaringer fra udviklingen af GEpj antyder, at modellering af arketyper (modsvarende GEpjs "submodellering") kræver ganske betydelig terminologisk indsigt og modelforståelse[upubliceret].

Jeg vil således umiddelbart stille mig skeptisk overfor Thomas Beales påstand om, at man kan adskille de to udviklingsteams[6]. På den anden side kan jeg ikke udelukke, at der kan udvikles værktøjer, som kan gøre denne adskillelse mere sandsynlig. Et minimalt krav må være, at værktøjet ikke tilbyder brugeren udtryksmuligheder, som sprænger rammerne for den foreliggende referencemodel.

3.4.4. Hvordan laver man trefoldig versionsstyring?

Jeg har i denne rapport bevidst forsøgt at minimere inddragelsen af klassifikationer og terminologi i overvejelserne omkring "dual-model"-konceptet. Men som man klart får indtryk af ved gennemgangen af openEHR's design-principper[9] og ved omtalen af [ADL](#)[17] - se Figur 14 - så er klassifikationer og terminologier en væsentlig ingrediens i et system

baseret på "dual-model"-metoden. Et sådant system baseres således på to modeller og mindst én terminologi.

At skitsere et versionsstyringssystem for en sådan treenighed ligger langt hinsides denne rapports ambitionsniveau, men det vil være på sin plads at fremhæve nogle centrale afhængigheder:

- Referencemodel, arketype og terminologi har mange og væsentlige gensidige afhængigheder[9]. Det forekommer indenfor et givent system af væsentlighed at anskue disse komponenters interaktion som et samlet hele.
- Referencemodellen ligger til grund for arketypen[6]. Ændringer i referencemodellen kræver derfor tilsvarende ændringer i samtlige berørte arketyper.
- Terminologien ligger til grund for arketypen[17] - hvis et værdisæt indskrænkes eller udvides, vil arketypen ikke nødvendigvis kunne forholde sig dertil. Ændringer i terminologien kræver derfor tilsvarende ændringer i samtlige berørte arketyper.

4. Har arketyper en plads i GEpj?

| | |
|---|-----------|
| 4.1. HVAD ER GEPJ? | 47 |
| 4.1.1. Niveauerne..... | 47 |
| 4.1.2. Dokumentations- eller implementeringsforskrift?..... | 48 |
| 4.1.3. Single- eller dual-model? | 49 |
| 4.1.4. Hvor generisk er "generisk"?..... | 50 |
| 4.2. ARKETYPER I GEPJ - NUTID ELLER FREMTID? | 52 |
| 4.2.1. Findes der allerede arketyper i GEpj? | 52 |
| 4.2.2. Er GEpj's referencemodel velegnet til arketypermodellering?..... | 53 |
| 4.3. GEPJ MED ARKETYPER - KONSEKVENSERNE | 54 |
| 4.3.1. Ændring af referencemodel?..... | 54 |
| 4.3.2. Ændring af implementeringsstrategi? | 55 |
| 4.3.3. Ændring af dokumentationsstrategi? | 55 |
| 4.3.4. Valg eller vej? | 56 |

4.1. Hvad er GEpj?

GEpj er Sundhedsstyrelsens Grundstruktur for Elektronisk Patientjournal[3], som af Sundhedsstyrelsen præsenteres som

[3]: ... en national specifikation af det kliniske indhold i en elektronisk patientjournal [EPJ]. Specifikationens ramme er den kliniske proces. G-EPJ specificerer krav til struktur, sammenhæng, omfang og formalisering af data i EPJ i ét sammenhængende, tværfagligt, tværsektorielt, problemorienteret og forløbsbaseret dokumentationssystem.

GEpj har tidligere haft andre formål og andre udformninger[3]. Når der i dette projekt henvises til detaljer i GEpj, drejer det sig - med mindre andet fremgår - om GEpj-2.0 pakke 20040416 ([25]), som blev publiceret i april 2004.

4.1.1. Niveauerne

GEpj er beskrevet i to forskellige abstraktionsniveauer - begrebsniveauet og det datalogiske niveau[26].

På begrebsniveauet tager GEpj sit udgangspunkt i en metafor, som beskriver en generelt problemløsende model[27] med begreber som problem, mål, plan og resultat (se Figur 13). Fra dette meget generiske

niveau sker der en (lille) konkretisering, idet GEpj indskrænker sig til at beskæftige sig med disse begreber indenfor det sundhedsfaglige område ved at:

[27 pg 5]: ... anskue klinisk proces som den sundhedsfaglige anvendelse af den generelle problemløsende proces.

GEpj fremstår herefter som en stadig meget generisk model med begreber som sundhedstilstand, sundhedsaktivitet og interventionsresultat - beregnet på formidling til sundhedsprofessionelle og lægfolk[26].

På det datalogiske niveau konkretiseres denne model på en måde, så den er egnet til formidling til leverandører og systemudviklere[26,28]. Dette omfatter en struktureret repræsentation udformet i UML ([29]).

UML-dokumentationen består af en række pakker med flg indhold:

- **Data types** - anvendte datatyper.
- **Grundstruktur** - klasser og usecases, som beskriver de overordnede (generiske) informationselementer og funktionaliteter.
- **Fejlpakken** - klasser og usecases vedrørende retning af fejl.
- **Indberetpakken** - klasser og usecases vedrørende indberetning af oplysninger til centrale sundhedsmyndigheder
- **Kontaktpakken** - klasser og usecases vedrørende tilstedeværelse.
- **Medicinpakken** - klasser og usecases vedrørende omgang med lægemidler.

En del af GEpj er beskrevet som "den logiske model"[28]. Denne del af GEpj vil - af hensyn til læseligheden - i resten af denne rapport blive benævnt GEpj's referencemodel.

4.1.2. Dokumentations- eller implementeringsforskrift?

I den samlede dokumentation, som udgør GEpj ([25]), har Sundhedsstyrelsen været tæt på både datastruktur og "glasplade" - der er specificeret en omfattende logisk model, et stort XML-schema, og visse steder har man specificeret arbejdsgange og anvendt illustrationer, som viser typiske skærm-komponenter.

Det er i denne forbindelse værd at bemærke, at GEpj

[3]: ... er ikke en specifikation af en journalopbygning, men en specifikation af krav til klinisk dokumentation i EPJ.

- dvs dokumentationen beskriver ikke, hvordan et system skal bygges eller se ud, men hvordan klinisk dokumentation skal hænge sammen.

Andetsteds udtrykkes dette i større detalje om GEpj's logiske model:

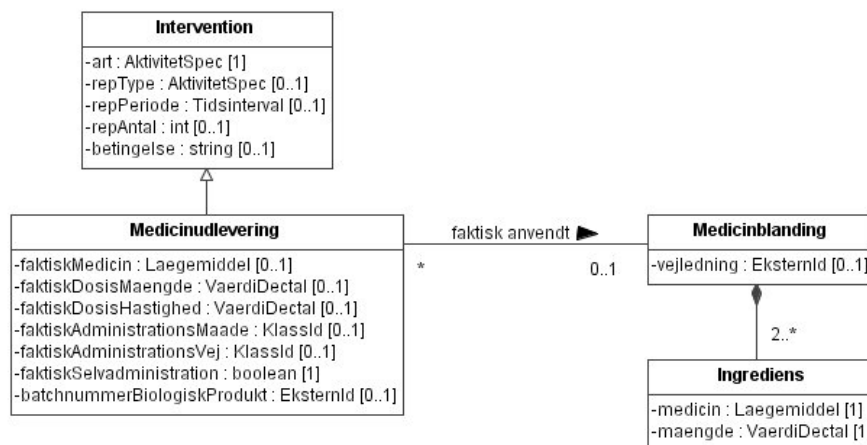
[28 pg 3]: ... at det ikke er krav, at et G-EPJ system har implementeret netop disse objekter eller lagre sine data i netop den struktur som GEpj's klassediagrammer foreskriver. Den logiske model fungerer som et sprog, som benyttes i resten af den datalogiske dokumentation ...

Jeg fortolker det således, at GEpj er ment som en forskrift for klinisk dokumentation i GEpj-baserede systemer, og at udviklere stilles frit med hensyn til fysisk implementering[28].

Som eksempel herpå bemærkes, at uagtet at der i GEpj-dokumentationen ikke findes arketyper (afsnit 4.2.1) bliver GEpj-lignende modeller implementeret ved hjælp af "dual-model"-metodik på fx Århus Amts DOM-plattform[personlig meddelelse].

4.1.3. Single- eller dual-model?

Hvis man anskuer pakkerne i UML-dokumentationen ([29]), står det ret hurtigt klart, at de er lavet på forskelligt abstraktionsniveau. Medens der i pakken Grundstruktur helt overvejende modelleres generiske begreber, som modsvarer begrebsniveauet ([27]), finder man i Indberetpakken, Kontaktpakken og Medicinpakken modellering af nogle helt domænespecifikke begreber. Disse er fortrinsvis udtrykt som specialiseringer af tilsvarende generiske begreber.



Figur 18. Udsnit fra GEpj's referencemodel (efter [29])

På Figur 18 er fx vist et udsnit fra GEpj's referencemodel. Her er klassen Intervention (som udtrykker det generiske begreb "sundhedsaktivitet"¹) forsynet med en domænespecifik specialisering - Medicinudlevering - med tilhørende domænespecifikke klasser Medicinblanding og Ingrediens.

Måden, hvorpå de domænespecifikke begreber er "submodelleret", vidner meget tydeligt om en tilgang, som nøje modsvarer den af Thomas Beale beskrevne "single-model"-metode[8]: én stor model, ét stort schema, klasser i forskelligt abstraktionsniveau, eksplicite attributter og relationer for de forskellige domænespecifikke begreber.

4.1.4. Hvor generisk er "generisk"?

Som anført i afsnit 4.1.1 udtrykker GEpj's referencemodel nogle generiske begreber anført i GEpj's begrebsniveau. Men hvor disse begreber i begrebsniveauet er meget generiske, synes de at "tabe højde", når de modelleres i det datalogiske niveau. Jeg ser dette som en konsekvens af den bagvedliggende "single-model"-metode (afsnit 4.1.3): det er i udviklingen ikke tilsræbt at afbilde generiske begreber som sådan, men i stedet at afbilde en fællesnævner for (kommende) domænespecifikke specialiseringer af disse begreber.

¹ Klassen Intervention udtrykker strengt taget kun de statiske dele af begrebet sundhedstilstand. Visse dele af begrebet - hvem, hvor, hvornår - udtrykkes i associerede klasser af typen Interventionsstatus[28 pg 10].

Resultatet bliver, at referencemodellen ikke i sig selv kan udtrykke instanser af de kliniske begreber eller - med andre ord - GEpj kan ikke rumme reel klinisk dokumentation før der foreligger de nødvendige specialiseringer af generiske begreber. Dette fænomen kan eksemplificeres:

Eksempel 3. Medicinudlevering uden specialisering af Intervention

På Figur 18 er vist referencemodellens klasse Intervention, der som anført udtrykker centrale dele af det generiske begreb sundhedsaktivitet. Det er ganske enkelt ikke muligt i denne klasse at udtrykke oplysninger af typen udleveret medicin, udleveret mængde, anvendt administrationsform etc. Man har bevidst valgt at udtrykke disse oplysninger gennem særlige klasser - som vist på figuren.

(Slut eksempel 3)

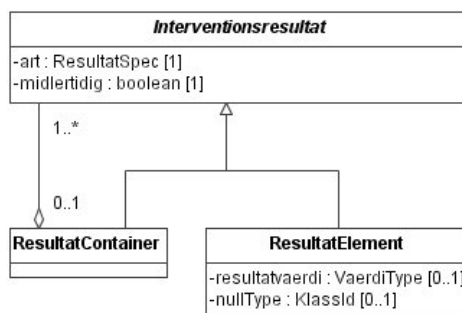
Formodentlig er det netop dette fænomen, som EpjObservatoriet i 2002 sigtede til med bemærkningen:

[1 pg 71]: ... der ligger fortsat en stor opgave med at identificere og detailspecificere nogle flere områder i GEpj på samme niveau som medicinering og røntgenområdet.

Hvis man ved udviklingen af GEpj i stedet havde tilstræbt at udtrykke generiske begreber på en generisk måde, kunne man stå med den modsatte situation - at referencemodellen kunne udtrykke enhver tænkelig instantiering af en række kliniske begreber. En sådan referencemodel ville have behov for begrænsning for at rumme klinisk anvendelig dokumentation. Et af GEpj's generiske begreber er modelleret på denne måde:

Eksempel 4. Modellering af resultat

I GEpj's referencemodel afbildes det generiske begreb resultat (opfattet som udkomme af aktivitet) som vist på Figur 19:



Figur 19. GEpj's modellering af resultat (efter [29])

Det generiske begreb resultat afbildes i klasserne Interventionsresultat, ResultatContainer og ResultatElement - se Figur 19. Denne modellering tillader at udtrykke information i vilkårlig dybde - det er således muligt at udtrykke et simpelt resultat af blodanalyse, et komplekst mikrobiologisk svar, resultatet af en røntgenundersøgelse etc.

(Slut eksempel 4)

4.2. Arketyper i GEpj - nutid eller fremtid?

4.2.1. Findes der allerede arketyper i GEpj?

Begrebet arketype (eller anden struktureret form for begrænsning af instantiering af referencemodellen) er ikke nævnt i GEpj[25]. Man kunne på den baggrund antage, at der ikke er anvendt arketyper ved udviklingen af GEpj.

Når spørgsmålet alligevel rejses, skyldes det, at EpjObservatoriet - da de i oktober 2002 udgav statusrapport - anførte, at GEpj's model for medicinering var en kandidat til en arketype[1]. Denne tilkendegivelse blev fulgt op med et forsøg på at udtrykke den daværende GEpj's medicinmodul ved hjælp af et arketypeværktøj udviklet af OpenEHR og anskuet i forhold til en referencemodel fra CEN/TC251[1 pg 49]. Forsøget var lidet illustrativt, og det er vanskeligt at opfatte det som overensstemmende med den i forrige kapitel anførte [opfattelse af arketyper](#).

Det kan derfor være relevant - med den her anvendte opfattelse af arketyper - at se, om den nuværende afspejling af omgang med lægemidler er gjort til genstand for arketypemodellering. En kort tilbagevenden til Eksempel 3 giver anledning til et ganske entydigt svar: Den nuværende modellering af fx Medicinudlevering udtrykker ikke en begrænsning, men en **udvidelse** af referencemodellens afbildning af det generiske begreb Sundhedsaktivitet. Der kan således ikke være tale om arketypemodellering - jvf [definitionen på arketype](#).

Dette er - som anført i de foregående afsnit - helt forventeligt: GEpj er udviklet efter "single-model"-metoden, og man har ved "sub-modelleringen" ikke tilstræbt at begrænse generiske begreber, men tværtimod at udvide dem gennem specialisering.

4.2.2. Er GEpj's referencemodel velegnet til arketypemodellering?

I det tidligere Eksempel 3 er det demonstreret, at modelleringen af det generiske begreb sundhedsaktivitet ikke tillader at man udtrykker en Medicinudlevering. Konsekvensen er, at man med den givne referencemodel ikke kan udtrykke fx medicinudlevering ved hjælp af begrænsning - der er simpelthen for lidt at begrænse, og arketyfificering har derfor ikke har megen mening. Sagt anderledes: Med GEpj's nuværende afbildning af sundhedsaktivitet, er denne del af referencemodellen ikke velegnet til arketypemodellering.

Anderledes forholder det sig med GEpjs modellering af det generiske begreb resultat, som det er demonstreret i Eksempel 4. Denne modellering tillader at udtrykke information i vilkårlig dybde og er dermed langt bedre egnet til at blive begrænset af arketyfificering.

Der findes således i GEpj både begreber, hvis modellering skønnes uegnet som underlag for arketypemodellering og begreber, hvis modellering skønnes velegnet hertil.

4.3. GEpj med arketyper - konsekvenserne

Det er i afsnit 4.1.3 godtgjort, at GEpj er udviklet med "single-model"-metode. Det er i afsnit 3.2 sandsynliggjort, at "single-model"-udvikling indebærer en række uhensigtsmæssigheder. Det er desuden i afsnit 4.1.4 påvist, at reel anvendelse af GEpj kræver en (kommende) submodellering.

Jeg kan på den baggrund forestille mig, at Sundhedsstyrelsen kunne have stor interesse i at skifte til "dual-model"-metode - dels for at opnå en mindre og mere stabil referencemodel, dels for at kunne flytte dele af udviklingsopgaven "ud af huset" - fx til Det Nationale Begrebsråd for Sundhedsvæsnet (NBS).

Hvis man forestiller sig, at Sundhedsstyrelsen ændrer sin udviklingsstrategi fra den nuværende "single-model"-metode til "dual-model"-metode - hvilke konsekvenser vil det få?

4.3.1. Ændring af referencemodel?

Det er i afsnit 4.2.2 godtgjort, at der i den nuværende referencemodel er modelleret generiske begreber på flere måder - hvoraf nogle er velegnede og nogle uegnede som underlag for arketypermodellering. Hvis Sundhedsstyrelsen ændrer udviklingsstrategi, må jeg forudse at der vil ske betydelige ændringer i referencemodellen med henblik på at afbilde generiske begreber på en mere generisk måde, jvf afsnit 4.1.4.

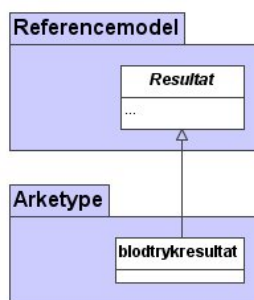
En sådan ændring af referencemodellen vil selvfølgelig umiddelbart være til besvær for de leverandører, som udvikler GEpj-baserede IT-systemer.

Men på sigt må det - også for leverandørerne - betragtes som en fordel, fordi det (afsnit 3.3.1) må antages at føre til en mindre og mere stabil referencemodel og fordi det er betingelsen for at høste de fordele, som er forbundet med at implementere en "dual-model"-applikation (afsnit 3.3.1).

4.3.2. Ændring af implementeringsstrategi?

Som anført i afsnit 4.1.2 er GEpj en forskrift for klinisk dokumentation i GEpj-baserede systemer - GEpj udsiger således ikke noget om, hvordan et GEpj-baseret IT-system skal implementeres.

Det er tidligere påpeget (afsnit 4.1.2), at en GEpj **uden** arketyper kan implementeres med "dual-model"-metode. På samme måde kan en GEpj **med** arketyper implementeres med "single-model"-metode:



Figur 20. "Single-model"-approach til arketyper

På Figur 20 - som skal ses i sammenhæng med reference- og arketypermodellen fra Eksempel 2 - er illustreret, hvordan en arketype kan opfattes som en specialisering af en klasse i referencemodellen. Det er således muligt at anskue et "dual-model"-system som om det var et (omend ganske granuleret) "single-model"-system.

Set på denne baggrund må jeg derfor konkludere, at en GEpj med arketyper ikke i sig selv pålægger nogen en forpligtigelse til at implementere arketyper. Men det åbner en mulighed for at høste fordelene ved "dual-model"-implementering (afsnit 3.3.1).

4.3.3. Ændring af dokumentationsstrategi?

Hvis Sundhedsstyrelsen ændrer sin udviklingsstrategi må jeg forvente at dette straks afspejler sig i en ændret dokumentationsstrategi - udover dokumentation af referencemodellen må der nødvendigvis også skulle dokumenteres arketyper. Dette rejser spørgsmål både om sprog og om kilde:

Som anført i afsnit 3.3.2 foreligger der et udkast til et sprog ([ADL](#)) til at udtrykke arketyper. Det foreligger kun i en arbejdsversion og må indtil videre betragtes som ustabil. En anden mulighed er at udtrykke arketyper i prosa og UML - dvs på samme måde som dokumentationen af referencemodellen.

Hvis "dual-model"-metoden indfrier løfterne med hensyn til at skille udviklingsteamet i et datalogisk og et ontologisk (se afsnittene 3.3.1 og 3.4.3) kan jeg forestille mig, at GEpj's arketyper udvikles af - og publiceres fra - fx NBS eller et leverandør-forum.

4.3.4. Valg eller vej?

Hvis Sundhedsstyrelsen skifter udviklingsstrategi for GEpj - skal skiftet så ske "i ét hug" eller kan man gå gradvis til værks? Svaret på dette spørgsmål ligger ved nærlæsning af afsnit 4.2.2 lige for: I den nuværende GEpj eksisterer der - side om side - generiske begreber, som mangler submodellering og generiske begreber, som mangler begrænsning.

GEpj er således en model, hvor der co-eksisterer elementer af både "single-model" og "dual-model", og det er dermed givet, at Sundhedsstyrelsen gradvis kan ændre udviklingsstrategi - og faktisk nok er i fuld gang med at gøre det.

5. Konklusion og perspektivering

| | |
|--|----|
| 5.1. KONKLUSION | 57 |
| 5.1.1. Har "arketyper" en plads?..... | 57 |
| 5.1.2. Har arketyper en plads i GEpj?..... | 59 |
| 5.2. VEJEN FREMAD..... | 60 |

5.1. Konklusion

Konklusionerne fra denne rapport falder naturligt i to dele som svar på de centrale spørgsmål anført som overskrift i de to foregående kapitler:

- **Har "arketyper" en plads?** - forstået som: Hvad er det tilgrundliggende koncept, og har det en mening og en anvendelse?
- **Har arketyper en plads i GEpj?** - forstået som: Kan konceptet om arketyper anvendes i udviklingen af GEpj, og vil det være formålstjenligt at anvende det der?

5.1.1. Har "arketyper" en plads?

I kapitel 3 er beskrevet det for arketyper grundliggende koncept: en metode til systemudvikling - ["dual-model"-metoden](#) - som er baseret på at beskrive systemet i to samtidige og gensidigt afhængige modeller, hvoraf arketype er den ene model.

Jeg finder det sandsynliggjort, at "dual-model"-metoden (læs: anvendelse af arketyper) kan indebære væsentlige fordele i udvikling og vedligeholdelse af sundhedsfaglige IT-systemer. Specielt bør nævnes (3.3.1):

- "Dual-model"-metoden synes at indebære et potentiale for fremtidssikring af IT-systemer
- "Dual-model"-metoden tilsigter at generalisere og minimere systemets tilgrundliggende referencemodel

- "Dual-model"-metoden synes at kunne reducere kravet om, at systemet er "komplet" på udviklingstidspunktet

idet det dog bemærkes, at

- Metodens potentiale for fremtidssikring synes at være afhængig af en helt pragmatisk opdeling af problemfeltets begreber - se afsnit 3.3.3

Jeg finder det i mindre grad sandsynliggjort, at

- udvikling af metodens to modeller kan varetages af separate udviklingsteams med forskellige kompetencer - se afsnit 3.4.3

Jeg konstaterer, at

- systemer baseret på "dual-model"-metoden har været realiseret, og metoden har således leveret proof-of-concept (3.4.1)

men jeg bemærker dog, at

- implemetering af en "dual-model"-applikation stiller krav om øget overhead til run-time-operationer (3.4.1).
- de foreliggende vidnesbyrd om implementerede "dual-model"-applikationer omfatter bemærkninger om manglende performance (3.4.2).

Desuden bemærkes, at

- jeg kan forestille mig problemer med i praksis at udtrykke og formidle arketyper - der foreligger ingen konvention om et foretrukket sprog (3.3.2).
- jeg kan forestille mig problemer med at versionsstyre "dual-model"-udvikling - herunder den gensidige afhængighed mellem referencemodel, arketype og terminologi (3.4.4).

Sammenfattende må jeg konkludere, at arketyper (læs: "dual-model"-metoden) **har** en plads som et meget spændende og visionært bud på en udviklingsmetode for sundhedsfaglige IT-systemer, at metoden har nogle betydelige og efterspurgte potentialer, men at den også præsenterer nogle udfordringer:

- "dual-model"-applikationer skal kunne fremvise acceptabel performance

- arketyper skal kunne formidles
- "dual-model"-systemer skal kunne versioneres

5.1.2. Har arketyper en plads i GEpj?

I kapitel 4 er det - med udgangspunkt i EpjObservatoriets beskrivelse af arketyper i GEpj[1] - søgt belyst, hvorvidt konceptet om arketyper (læs: ["dual-model"-metoden](#)) er anvendt eller kan anvendes i udviklingen af GEpj.

Jeg finder det godt gjort, at

- der foreligger i GEpj ikke anvendelse af arketyper (4.2.1)

Jeg finder det godt gjort, at

- GEpj er udviklet efter "single-model"-metoden (4.1.3)

omend

- GEpj indeholder mindst ét begreb (resultat), som gennem sin afbildning i referencemodellen synes at være velegnet til arketypermodellering (4.2.2).

Jeg finder det sandsynliggjort, at

- anvendelse af "dual-model"-metoden i udviklingen af GEpj kan være til fordel for både Sundhedsstyrelsen og leverandører (4.3)

men at det

- kræver afklaring og løsning af problemerne vedrørende formidling af arketyper (5.1.1)
- kræver afklaring og løsning af problemerne vedrørende versionsstyring (5.1.1)
- indebærer betydelige ændringer i referencemodellen (4.3.1)

Det bemærkes i denne forbindelse, at

- Sundhedsstyrelsens overgang til "dual-model"-metoden pålægger ikke i sig selv leverandørerne at implementere GEpj-baserede systemer som "dual-model"-systemer (4.3.2)
- Sundhedsstyrelsen kan gradvist overgå til "dual-model"-metode (4.3.4)

Sammenfattende må jeg konkludere, at arketyper (læs: "dual-model"-metoden) **kan have** en plads i GEpj, og at en sådan ændring af Sundhedsstyrelsens udviklingsmetode må anses at være til fordel både ved udvikling og implementering af sundhedsfaglige IT-systemer.

5.2. Vejen fremad...

I 2002 skrev Thomas Beale:

[8]: Executed correctly, a two-level modelling methodology stands to radically change the economics and quality of information systems. As far as is known, none of the existing software development methodologies takes a two-level approach...

I oktober samme år skrev EpjObservatoriet:

[1]: EPJ-Observatoriet foreslår, at der iværksættes et projekt for at:
- undersøge og teste om arketyper kan bruges som harmoniseringsværktøj
- vurdere værktøjer til dokumentation af arketyper

Nu - 1½ år efter - foreligger GEpj i en nyslået udgave([25]) uden anvendelse af arketyper, og der foreligger ingen afklaring af værktøjer til dokumentation af arketyper. Så - hvordan kommer man videre?

Som det fremgår af det anvendte kildemateriale, er "dual-model"-metoden ikke med bredde blevet udfordret - hverken praktisk eller teoretisk (jvf afsnit 2.2.5). De kommende år vil med sikkerhed bringe yderligere erfaringer med denne metode - både internationalt og forhåbentlig også her i landet.

I denne rapport konkluderes, at Sundhedsstyrelsen i sit arbejde med at udvikle GEpj kan have gavn af at anvende "dual-model"-metoden og at en sådan ændring af Sundhedsstyrelsens udviklingsmetode må anses at være til fordel både ved udvikling og implementering af sundhedsfaglige IT-systemer. Sundhedsstyrelsen kan foretage denne ændring af udviklingsmetode gradvist.

Men det kræver, at der sker en afklaring af nogle udeståender:

Arketyper skal kunne skrives og formidles

Der vil være behov for et værktøj, som - på grundlag af et givent stykke referencemodel - tillader et udvikler-team at udtrykke en arketype. Det bør undersøges, om det under OceanInformatics udviklede arketypeværktøj (http://www.oceaninformatics.biz/adl_downloads.html) er tilfredsstillende eller om der er brug for egenudvikling.

Output fra dette værktøj skal foreligge i et format, som er både humant og maskinelt læsbart. Det i slutningen af afsnit 3.3.2 nævnte sprog (ADL) er et oplagt bud på et sådant format. Jeg bemærker, at sproget endnu ikke er stabilt - spørgsmålet er, om den manglende stabilitet udgør et faktisk problem, så længe den centrale opgave er at formidle arketyper humant. Det vil være et langt større problem, når arketyper skal formidles maskinelt...

Alternativet til et sprog, som man skal udvikle, lokalt stabilisere eller blot acceptere ustabiliteten af, er at anvende et sprog, som kun er humant læsbart. Én mulighed ville være at udtrykke arketyper i pseudo-UML, som det er gjort i denne rapport's Figur 9.

"Dual-model"-systemer skal kunne versioneres

Jeg har i denne rapport undladt at forholde mig andet end helt overfladisk til problemstillingen, at medens GEpj aktuelt (kun!) skal versionsstyre én model og én klassifikation (3.4.4), så skal en GEpj baseret på "dual-model"-metode kunne versionsstyre to gensidigt afhængige modeller og mindst én klassifikation.

Inden Sundhedsstyrelsen ændrer udviklingsstrategi, bør der altså foreligge en plan for versionsstyring.

Modellering af arketyper bør out-sources

Som det vil være bekendt for deltagerne i GEPKA-projektet, laves GEpj af et meget lille team[personlig observation]. Som anført ovenfor vil en ændring af udviklingsstrategien kunne indebære flere fordele for Sundhedsstyrelsen, men det største potentiale ligger formodentlig i, hvis der kan oprettes separate teams til referencemodellering og til

arketypemodellering, og hvis arketype-teamet kan rekrutteres udenfor Sundhedsstyrelsen.

Der er tidligere stillet spørgsmålstegn ved, om dette faktisk er muligt (3.4.3), men som anført kan gode værktøjer gøre den udviste skepsis til skamme...

Oplagte teams til at indgå i arketypemodellering synes at være arbejdsgrupper fra NBS og videnskabelige selskaber, men jeg kan også forestille mig spændende bidrag fra teamet bag Århus-DOM.

Referencemodellen skal ændres

Dette punkt er jo en integreret del af at ændre udviklingsstrategien, som anført i afsnit 4.3.1. Grunden til, at det er nævnt her som et selvstændigt punkt er en konsekvens af det politiske klima, som GEpj aktuelt udvikles under: Med et krav om landsdækkende udbredelse af EPJ om 1½ år[4] og med multiple amtslige udbudsforretninger baseret på GEpj - så er der et betydeligt politisk pres for at holde referencemodellen stabil.

Jeg kan befrygte, at dette politiske pres som konsekvens får en ringere GEpj-udvikling.

Der er nok at tage fat på, og i morgen venter næppe...

6. Litteratur

- 1 Andersen SK, Nøhr C et al. EPJ-Observatoriet - Statusrapport 2002. Oktober 2002. [Online, PDF] <http://www.EPJ-Observatoriet.dk/publikationer/Statusrapport2002.pdf> (Verificeret 2004-05-08)
- 2 Bernstein K, Bruun-Rasmussen M. EPJ og integration. 2002. [Online, PDF] <http://www.epj-observatoriet.dk/konference2002/konferenceslides/KnutBernstein.pdf> (Verificeret 2004-05-08)
- 3 Sundhedsstyrelsen. Grundstruktur for Elektronisk Patientjournal (G-EPJ). [Online, HTML] <http://medinfo.dk/epj/gepj/> (Verificeret 2004-05-08)
- 4 Indenrigs- & Sundhedsministeriet. National IT-strategi for sundhedsvæsenet 2003-2007. Maj 2002. [Online, PDF] <http://www.im.dk/publikationer/itstrategi/itstrategi.pdf> (Verificeret 2004-05-08)
- 5 Nøhr C. Computer-Assisted Decision-Support - Better Health? 1991. Aalborg Universitetscenter, Aalborg.
- 6 Beale T. Archetypes - Constraint-based Domain Models for Futureproof Information Systems, rev 2.2.1. August 2001. [Online, PDF] <http://www.deepthought.com.au/it/archetypes/archetypes.pdf> (Verificeret 2004-05-08)
- 7 openEHR. The Dual Model approach. [Online, HTML] http://www.openehr.org/getting_started/t_dualModel.htm (Verificeret 2004-05-08)

- 8 Beale T. Archetypes: Constraint-based Domain Models for Future-proof Information Systems (for OOPSLA 2002 workshop on behavioural semantics). 2002. [Online, PDF]
http://www.deepthought.com.au/it/archetypes/archetypes_new.pdf (Verificeret 2004-05-08)
- 9 Beale T, Goodchild A, Heard S. Design Principles for the EHR, revision 2.4. April 2002. [Online, PDF]
http://www.openehr.org/downloads/design_principles_2_4.pdf (Verificeret 2004-04-30¹)
- 10 Karlsen A, Galster G et al. Sundhedsvæsnets begrebsbase - den medicinske babelfisk. Maj 2002. [Online, HTML]
<http://galster.dk/gert/docs/MIproj2/> (Verificeret 2004-05-08)
- 11 Rector AL. Clinical Terminology: Why Is It So Hard? 2001. Yearbook of Medical Informatics 2001.
- 12 Sundhedsstyrelsen. G-EPJ prototype og klinisk afprøvning (GEPKA) Projektbeskrivelse. Januar 2003. [Online, DOC]
http://medinfo.dk/epj/proj/gepka/projbeskr_1_0.doc (Verificeret 2004-05-08)
- 13 Biering-Sørensen S, Hansen FO et al. Håndbog i struktureret program-udvikling. 1996. Teknisk Forlag, København.
- 14 Health Level Seven. HL7 Reference Information Model, version: V 02-02 (11/22/2003). November 2003. [Online, HTML]
<http://www.hl7.org/library/data-model/RIM/C30202/rim.htm> (Verificeret 2004-05-08)
- 15 CEN/TC 251. EHRCOM prEN 13606-1 - 2nd working draft. Marts 2004. [Online, PDF]

¹ OpenEHR lavede sit Internet-site om den 1/5-2004. Det har siden været umuligt at finde referencen [9] og webmaster svarer ikke på henvendelser. Frem til eksamen vil en kopi være tilgængelig på adressen
http://galster.dk/gert/docs/siproj3/F05_design_principles_2_4.pdf

- http://www.centc251.org/TCMeet/doclist/TCdoc04/N04-012prEN13606-1_2WD.pdf (Verificeret 2004-05-08)
- 16** Beale T, Heard S. A Shared Archetype and Template Language Part I. Maj 2003. [Online, DOC]
http://www.deepthought.com.au/health/archetypes/archetype_language_1v0.3.doc (Verificeret 2004-05-08)
- 17** Beale T, Heard S. A Shared Archetype and Template Language Part II. 2003. [Online, DOC]
http://www.deepthought.com.au/health/archetypes/archetype_language_2v0.7.doc (Verificeret 2004-05-08)
- 18** openEHR. Archetypes and Templates FAQ. [Online, HTML]
http://www.openehr.org/FAQs/t_archetypes_FAQ.htm
(Verificeret 2004-05-15)
- 19** NHS Information Authority. Electronic Record Development and Implementation Programme Update March 2003. Marts 2003. [Online, PDF]
<http://www.nhsia.nhs.uk/erdip/pages/publications/ERDIPUpdateMar03.pdf> (Verificeret 2004-05-08)
- 20** Fearon A, Parnell S. ERDIP Project Evaluation Report, Volume 1: Summary. November 2002. [Online, PDF]
[http://www.nhsia.nhs.uk/erdip/pages/demonstrator/devo/devon_\(32\).pdf](http://www.nhsia.nhs.uk/erdip/pages/demonstrator/devo/devon_(32).pdf) (Verificeret 2004-05-08)
- 21** Avenir, Systematic, Århus Amt. Domæne objektmodel. November 2000. [Online, PDF] http://www.aaa.dk/aaa/dom_2.pdf
(Verificeret 2004-05-08)
- 22** EPJ-Observatoriet. Vurdering af Århus Amts EPJ, Version 1.0. April 2004. [Online, PDF]
<http://www.aaa.dk/dagsor/sh/230404/bilag/HBN-rap-1-2.pdf>
(Verificeret 2004-05-15)

- 23** Projektledelsen for Århus Amts EPJ. Kommentarer til EPJ-Observatoriets rapport "Vurdering af Århus Amts EPJ". April 2004. [Online, PDF]
<http://www.aaa.dk/dagsor/sh/230404/bilag/HBN-1-3.pdf>
(Verificeret 2004-05-25)
- 24** Parnell S, Fearon A. ERDIP Project Evaluation Report, Volume 2: Proof of Concept. November 2002. [Online, PDF]
[http://www.nhsia.nhs.uk/erdip/pages/demonstrator/devo/devon_\(33\).pdf](http://www.nhsia.nhs.uk/erdip/pages/demonstrator/devo/devon_(33).pdf) (Verificeret 2004-05-08)
- 25** Sundhedsstyrelsen. G-EPJ, udgivelse Version 2.0 pakke 20040416. April 2004. [Online, HTML]
http://medinfo.dk/epj/gepj/020_20040416/index.html
(Verificeret 2004-05-08)
- 26** Sundhedsstyrelsen. GEpj udviklingsniveauer. April 2004. [Online, DOC]
http://medinfo.dk/epj/gepj/020_20040416/GEpjPk_Niveau_020_.doc (Verificeret 2004-05-08)
- 27** Sundhedsstyrelsen. Beskrivelse af GEpj - på begrebsniveau. April 2004. [Online, DOC]
http://medinfo.dk/epj/gepj/020_20040416/GEpjBn_Beskriv_020_.doc (Verificeret 2004-05-08)
- 28** Sundhedsstyrelsen. Beskrivelse af GEpj - på datalogisk niveau. April 2004. [Online, DOC]
http://medinfo.dk/epj/gepj/020_20040416/GEpjDn_Beskriv_020_.doc (Verificeret 2004-05-08)
- 29** Sundhedsstyrelsen. GEpjDn Uml 2.0 20040416. April 2004. [Online, HTML]
http://medinfo.dk/epj/gepj/020_20040416/RapDn/index.html
(Verificeret 2004-05-08)