

- Kommercielle løsninger tillader ikke umiddelbart udvikling af nye regelsæt, som er baseret på åbne standarder og ikke er direkte afhængige af de enkelte softwareimplementeringer. [47, 19].
- Der mangler løsninger for prægranskning [5].
- Der kræves en fælles process mellem regel- eller lovforfattere og de praktikere der skal overholde reglerne. Denne proces skal resultere i en klar definition af regler ift. lov, bygbarhed osv. som kan fortolkes af et eller flere kontrolværktøjer.
- Forskning, inddragelse af interessenter og udvikling af regler sammen med et fælles sprog er nødvendige skridt for at løse de nuværende ARC-problemer i branchen. Produktudvikling og skalering er de efterfølgende trin for at gennemføre en velfungerende løsning [5].
- Reglerne bør tilpasses efter projektstadiet (LOD) og det kræver en fleksibel og lettilgængelig tilgang til oprettelse af regler [5, 36].
- Der er kun behov for den relevante delmængde af datamodellen for at tjekke fagspecifikke regler [36, 24].
- ARC kræver en klar forståelse af semantikken i de undersøgte data og i de regler der ønskes anvendt [36, 47].
- Maskinlæsbare regler bør kunne forstås af forfatterne af regulativer [19].
- Regelbaser bør være uafhængige af Computer Aided Design (CAD)-implementeringer [19].
- Mangel på fælles åbne standarder for specifikation af regler og ARC-håndtering [19, 5, 36].
- Kløften mellem de data, der skal kontrolleres og reglernes ordlyd kræver at ARC-applikationer er i stand til at finde implicitte data ud fra de oplysninger, der eksplícit er til rådighed [36] (Fig. 1).
- Modeltjek bør integreres med modelskabelsesprocessen for at sikre anvendelighed af regeltjekkerne [19].
- De første to klasser af regler (der kræver eksplicitte og afledte data) er relativt enkle at gennemføre, og alligevel har de et enormt potentiale til at bidrage til at forbedre kvaliteten af bygningsmodeller [36].
- Der findes ikke nogen åbent anvendelig løsning, der kan integrere regler med den kompleksitet og de krav, der gælder for klasse 3 og 4 (regler der kræver udvidet datastruktur og bevisførelse) [36]. SMC er i stand til at bruge klasse 3-regler, men det kræver en stor ekspertise i det der skal indarbejdes i programkoden (ufleksibelt) [21].
- Semantiske webteknologier giver en større fleksibilitet, og tillader et skalerbart udviklingsmiljø i en standardiseret datamodel (RDF). Det er fordelagtigt ved udvikling af nye regelsæt og funktionalitet [47].
- BimSPARQL tillader at beregne implicitte data og hvis koblet med en geometri-engine kan man selv integrere regler af klasse 3 og 4 [47].

## 4 Åbne standarder

Flere studier i forgående afsnit understreger vigtigheden af åbne standarder til at kunne beskrive projektdatasættet (BIM-modellen) såvel som de regler som

dette datasæt skal valideres op imod gennem ARC. I dette afsnit undersøges nogle af de åbne standarder der er tilgængelige inden for klassifikation og dataudveksling i byggebranchen, og det berøres hvordan disse kan sammensmelte ved brug af semantiske webteknologier og linked data.

## Klassifikation

Vigtigheden af åbne standarder har rødder længere tilbage end digitaliseringen af byggeriet, men ikke desto mindre er den en fundamental nødvendighed for at sidstnævnte kan lykkes [1].

## Klassifikation af objekter

En ”klasse“ og det at ”klassificere“ bliver af Ekholm & Häggström [11] defineret som:

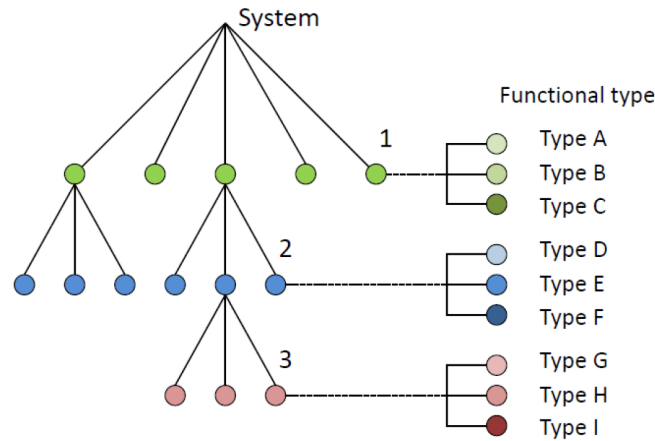
**Klasse:** *“et konceptuelt begreb, der henviser til en samling information-subjekter med en eller flere egenskaber til fælles.”*

**Klassificere:** *“at opdele en samling objekter i indbyrdes adskilte sæt eller klasser.”*

Klassifikation organiserer de mest generelle klasser i de højere niveauer (root levels). Derefter arrangeres mere specifikke klasser på de lavere niveauer. Ethvert punkt i dette hieraki er en specialisering af sine overklasser og enhver overklasse er en generalisering af dennes underklasser [20]. Udover den overordnede hierakiske indeling af klasser er der også klassifikationssystemer som benytter sig af underinddelinger som det eksempelvis er vist med funktionelle typer i figur 5 [11]. Ethvert klassifikationssystem er bygget ud fra et særligt formål og har derfor sin egen måde at definere det overordnede hieraki samt eventuelle underhierakier. Det gør det umiddelbart svært at mappe direkte imellem dem [1]. Det mangfoldige landskab af klassifikationssystemer (Samarbetskommitten for Byggnadsfragor (SfB), dansk byggeklassifikation (DBK), CCS/CCI, BIM7AA, Forvaltnings Klassifikation, CoClass, Uniclass, OmniClass, MasterFormat, Unifomat, Freeclass m.fl.) vidner om et stort behov for entydigt at kunne klassificere bygningsobjekter. Samtidig udstiller det en uheldig tendens til at man ofte starter forfra og kun i ringe grad bygger videre på hvad der i forvejen eksisterer. Denne tendens har til dels været funderet i begrænsninger i den tilgængelige teknologi, men med webteknologier har vi muligvis et værktøj til at overkomme denne udfordring. Det vender vi tilbage til i et senere afsnit.

## Klassifikation af rum

Særligt når det gælder adgangsveje, indeklimakrav mv. er det nødvendigt med en entydig betegnelse for en rumtype. I et studie fra 2012 undersøger Lee et al. [22] en metode hvor de automatisk forsøger at mappe rumnavne op imod en



**Fig. 5.** En klassificeringsstruktur med bestanddele og deres funktionelle typer. Tallene 1-3 angiver niveauer af part-helheds-relationer [11].

database med entydige rumtyper, netop for at kunne udføre ARC på modellerne. Her anbefales det at hvis der skal udføres en 100 % mapping af rumnavnene, så bør der som minimum gives nogle retningslinjer. En anden tilgang er simpelthen at klassificere rummene lige som det gøres for bygningsdele. Sådan et klassifikationssystem har både CCS og Real Estate Core (REC)<sup>2</sup> bud på.

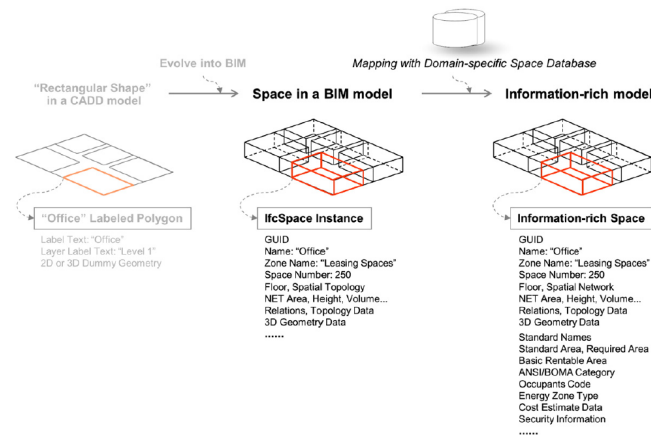
### Klassifikation af egenskaber

Klassifikation af objekter hører uundgåeligt sammen med specifikation af de egenskaber, som er associeret med objektet. Der er stor forskel på detaljeringsgraden af objektklasser. Som Rasmussen skriver i sin Ph.d.-afhandling benytter CCS en mere overordnet klassificering i modsætning til OmniClass som har meget detaljerede klasser. Eksempelvis defineres et 'indvendigt ikke-oplukkeligt vindue' som '21-03 10 20 20'. I CCS er dette blot et vindue ('QQA') med to bolske egenskaber 'indvendigt' og 'oplukkeligt' tilknyttet [32]. Lige som det er nødvendigt med en entydig klassifikation af objekter, er det ligeledes vigtigt med en entydig navngivning af egenskaber.

### Normer og standarder

Mange af de egenskaber som benyttes i diverse ingeniørdiscipliner, er entydigt beskrevet i normer og standarder, der udgives af anerkendte organisationer som bl.a. Dansk Standard (DS) (nationalt), European Committee for Standardization (CEN) (europæisk), American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) (nordamerikansk) og International Organi-

<sup>2</sup> <https://doc.realestatecore.io/3.2/building.html>



**Fig. 6.** Oversættelse af simpel CAD-geometri til IfcSpace-instanser til informationsrige rum[22].

zation for Standardization (ISO) (internationalt). Eksempelvis indeholder standarden for beregning af bygningers varmetab (DS 418) indledningsvis en beskrivelse af anvendte begreber og en angivelse af symboler og enheder. Her defineres blandt andet begreber som det 'dimensionerende varmetab' og en 'energiramme'. De definitioner der angives i standarder fra DS, CEN, ISO osv.

### IFC, Property sets og bsDD

BuildingSMART, som står bag mange standardiseringsinitiativer ift. digitalisering af byggebranchen, er nok mest kendt for IFC filformatet. IFC-datamodellen beskriver mere dybdegående i næste afsnit, men her vil vi først komme omkring den klassifikation af bygningsobjekter i bestemt hieraki (se figur 7), som er indeholdt i IFC. Derudover beskriver IFC nogle faste egenskaber som kan suppleres med egenskaber beskrevet i såkaldte property sets. Eksempelvis beskriver IFC at en væg kan have geometriske egenskaber som en *NominalLength* og et *NetFootprintArea*<sup>3</sup>. Property sets er typisk målrettet et særligt formål, og er knyttet til en specifik objekttype. For vægge findes for eksempel den generelle 'Pset\_WallCommon', som beskriver nogle standardegenskaber som er typiske for vægge. Eksempelvis hvor vidt den sidder udendørs (*isExternal*<sup>4</sup>) og brandklasse (*FireRating*<sup>5</sup>). Det er muligt at definere sine egne property sets, og med buildingSMART Data Dictionary (bsDD), som også er et buildingSMART-initiativ er det muligt at publicere dem. bsDD indeholder egenskaber og objektklasser fra en del nationale standarder, klassifikationssystemer og endda projekt- og fir-

<sup>3</sup> <https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/FINAL/HTML/ifcsharedbldgelements/lexical/ifcwall>.

<sup>4</sup> [http://bsdd.buildingsmart.org/#concept/browse/3Yss80qXKHuO00025QrE\\$V](http://bsdd.buildingsmart.org/#concept/browse/3Yss80qXKHuO00025QrE$V)

<sup>5</sup> [http://bsdd.buildingsmart.org/#concept/browse/39EXg0qXKHuO00025QrE\\$V](http://bsdd.buildingsmart.org/#concept/browse/39EXg0qXKHuO00025QrE$V)

maspecifikke aftaler. Med den nye version 5<sup>6</sup> af bSDD som i skrivende stund er under udvikling<sup>7</sup>, er der understøttelse af den opdaterede ISO 12006-3, såkaldte *Object Type Libraries* og *Product Data Templates* og tilmed kan de publiceres som *Linked Data* (mere om dette senere)[44].

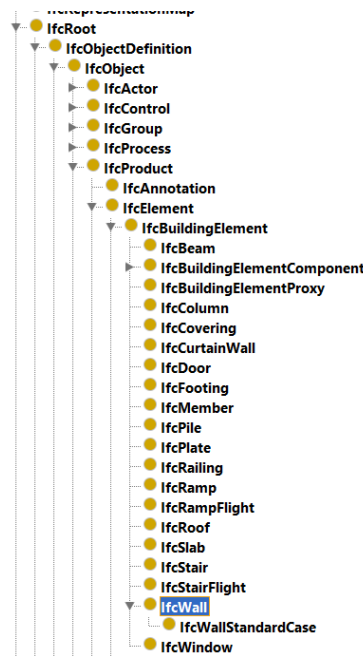


Fig. 7. Udpluk af IFC klasser for bygningselementer visualiseret med Protégé.

I Danmark har der været nationale initiativer til at beskrive egenskaber. Molio har tidligere beskrevet egenskabsdata ifm. CCS. Dette projekt er dog stoppet, og i stedet er det besluttet at benytte den kommercielle CoBuilder-platform til egenskabsdata. CoBuilder benytter lige som bSDD en intern datastruktur som understøtter ISO 23386 (beskrivelse, oprettelse og vedligeholdelse af egenskaber i sammenkoblede databøger) og ISO 23387 (beskrivelse af *Product Data Templates*). Formålet med alle disse løsninger er det samme: At angive en entydig betegnelse for en egenskab (navn eller globally unique identifier (GUID)), og herefter beskrive metadata om egenskaben såsom labels og beskrivelser på forskellige sprog samt eventuelt henvisning til målemetoder mv. Argumentet imod en kommerciel løsning er at den kun er tilgængelig bag en betalingsvæg. Argumentet for er, at der er en central instans, som sørger for at der opretholdes

<sup>6</sup> <https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/>

<sup>7</sup> <https://github.com/buildingSMART/bSDD/tree/master/2020%20prototype>

en fornuftig orden i egenskaberne - at der ikke oprettes redundante data og at der holdes en høj kvalitet med omfattende, opdaterede metadata.

En udfordring med såvel property sets og bSDD som med de kommercielle løsninger er, at de kun giver mulighed for at beskrive simple attributter - altså egenskaber hvis værdi er en simpel datatype som en tekststreng, en boolsk værdi eller et tal. Der er ikke mulighed for at beskrive relationer mellem objekter for eksempelvis at kunne beskrive at flere rum hører til samme lejlighed/hotelværelse eller at døren fungerer som en grænseflade med adgangskontrol (ADK)-begrænsning i den ene retning.

## IFC

I forrige afsnit blev IFC og de indeholdte klasser og egenskaber kort berørt. IFC er et åbent format til dataudveksling af BIM-data, og skemaet er enormt omfattende og rigt. Det indeholder som bekendt klasser og egenskaber, men også relationer mellem objekterne. Selvom IFC bygger på et rigt skema, har det dog sine begrænsninger. Af dem der typisk bliver påpeget i forskningen, er følgende typiske gengangere: [6, 48, 26, 14]

- IFC er baseret på EXPRESS/STEP og dermed en CWA. Det betyder at skemaet er nødt til at omfatte hele verden, idet det ikke er muligt sidenhen at udbygge det, og derfor er det uflexibelt i en byggebranche hvor der løbende opstår behov for at kunne beskrive nye produkter og relationer imellem dem. Udfordringen er delvist løst med klassifikationssystemer og property sets, men det er fx ikke muligt at udvide med nye relationer mellem objekter
- IFC er et udvekslingsformat, og det er i udgangspunktet read-only. Selvom software som SimpleBIM tillader at arbejde med en IFC-fil er mulighederne begrænsede, og round-trips tilbage til det BIM-værktøj som modellen blev skabt i, er generelt ikke mulig
- IFC mangler et query-sprog, og fordi EXPRESS/STEP kun har ringe udbredelse er der ikke mange software-integrationer. Derfor er det svært at hente data ud fra en IFC-fil
- Med MVDs er muligt at udveksle en delmængde af en model, men da dette i praksis er en filtreringsmekanisme der applikeres ved eksporten er det ikke at sammenligne med et query-sprog. Desuden er det komplekst at kombinere disse datasæt efterfølgende hvorfor dette ikke er normal praksis

Meget tyder på at IFC med tiden vil blive modulært<sup>8</sup>, og der er stor fokus på at øge tilgængeligheden ved at skifte EXPRESS/STEP ud med mere gængse udvekslingsformater som extensible markup language (XML) og JavaScript Object Notation (JSON). I erkendelse af at IFC aldrig kommer til at beskrive hele verden, og at bygningsdata ønskes kombineret med øvrige datasæt som eksempelvis geografiske, meteorologiske, og procesmæssige med hver deres egne standardiserede skemaer, har buildingSMART udgivet en version af IFC som understøtter det semantiske web kaldet ifcOWL.

<sup>8</sup> <https://forums.buildingsmart.org/t/ifc-modularisation-current-status-and-request-for-feedback/2054>

## ifcOWL

IfcOWL er en version af IFC-skemaet, som er beskrevet med webstandarderne RDF<sup>9</sup>, RDF Schema (RDFS)<sup>10</sup> og OWL<sup>11</sup>. I appendiks A er disse teknologier beskrevet mere dybdegående, men overordnet kan det fortælles at RDF er et framework til at beskrive og klassificere ressourcer (i bund og grund hvad som helst du kan tænke på - en væg, et vindue, en aktivitet, en aktør, en sensorobservation) samt deres egenskaber og indbyrdes relationer. RDF er ikke et dataformat, og det kan udveksles blandt andet som RDF/XML og JSON for Linked Data (JSON-LD). RDFS indeholder blandt andet begreber til at definere hierakier af klasser og egenskaber som det blev beskrevet tidligere i dette afsnit, og OWL giver mulighed for at beskrive logiske regler som eksempelvis kan beskrive at en egenskab er symmetrisk (gælder i begge retninger - fx "er i familie med") eller at en egenskab er det modsatte af en anden egenskab (fx "har barn"/"har forælder").

Med ifcOWL er IFC-skemaet blevet konverteret til en OWL ontologi. En ontologi er af Stunder (1998) [40] defineret som:

**Ontologi:** *"en formel, eksplicit specifikation af en delt konceptualisering."*

I delt menes at der er enighed inden for et fag/område (fx danske VVS-ingeniører eller tidsplanlæggere) og at den er gjort tilgængelig (publiceret online). Dermed kunne bygningsreglementet og diverse normer og standarder i princippet også formaliseres i ontologier.

En af de største styrker ved RDF er at det tilføjer det manglende skema til JSON, men på en måde der tillader at et datasæt beskrives med flere forskellige skemaer på en og samme tid. Det er altså ikke kun en IFC-bygning, men også en City Geography Markup Language (CityGML)-bygning, og lige som denne bygning kan have en geometrisk model beskrevet med EXPRESS/STEP, kan den også blot repræsenteres af et 2D-polygon, som kan vises på et kort (eksempelvis beskrevet med Well-Known Text (WKT)). Det er lidt af en game-changer, da det gør os i stand til at beskrive en bygningsmodel med flere forskellige skemaer - eksempelvis et skema som særligt fokuserer på bygningen (IFC), et med særlig fokus på Internet of Things (IoT)-devices, og et der beskriver den i en Geographic Information System (GIS)-kontekst. Da alle ressourcer i RDF navngives med webadresser er der endvidere mulighed for at datamodellen kan være distribueret på forskellige servere så dele af den eksempelvis kan ligge hos arkitekten, dele hos bygherre og dele hos de enkelte ingeniørdisipliner. Derudover er der mulighed for at henvise til og benytte offentligt tilgængelige datasæt med eksempelvis materiale- og bæredygtighedsdata [6, 30]. Perspektiverne i et decentraliseret BIM er mange, og der er også bred enighed om at det er der teknologien vil bære os hen i fremtiden - Succar betegner eksempelvis den højste modenhed

<sup>9</sup> <https://www.w3.org/TR/rdf11-primer/>

<sup>10</sup> <https://www.w3.org/TR/rdf-schema/>

<sup>11</sup> <https://www.w3.org/TR/owl2-primer/>

af BIM (som vi endnu ikke har opnået) som 'netværksbaseret integration' [41], og en lignende betegnelse findes i den britiske Publicly Available Specifications (PAS) 1192-2:2013.

En væsentlig fordel med ifcOWL er at den RDF-baserede datamodel giver mulighed for at udføre queries med et standardsprog kaldet SPARQL (se mere i appendiks A). Med BimSPARQL [48] beskrev Zhang et al. desuden en metode til at lave geometriske queries på en ifcOWL-model (se afsnit 2).

### Linked Building Data

IfcOWL løser desværre ikke problemet med at IFC er en meget omfattende datamodel, og konverteringen til RDF har betydet at der er en del levn fra det EXPRESS/STEP skema den kommer fra. Det gør den resulterende RDF-graf unødigt kompliceret at udføre queries på, og derfor er der først blevet arbejdet med at forsimple den [29], at berige den med genveje, som gør den lettere at tilgå [48] og at gøre den modular [42]. Sidenhen er der af World Wide Web Consortium (W3C)-gruppen for Linked Building Data (LBD) blevet foreslået en anden tilgang som bygger på en modular struktur med the Building Topology Ontology (BOT) som en kerne der udbygges med domænespecifikke udvidelser [33]. BOT har vist sig at være en fleksibel løsning som egner sig til ARC og beskrivelse af IoT-devices i kontekst af bygninger [8, 7, 13, 43]. BOT er endvidere alignet med en del ontologier som berører bygninger [34].

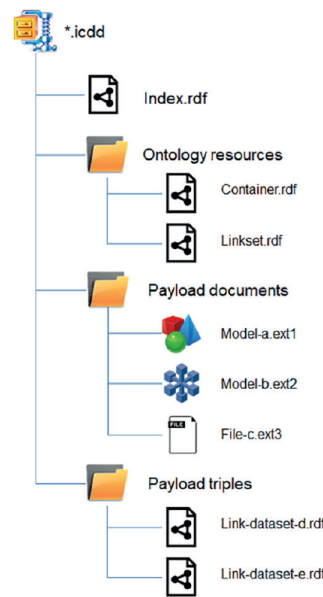
Da BOT kun giver rammen i RDF til at beskrive en bygnings topologiske opbygning, giver det desuden mulighed for at integrere flere klassifikationssystemer og derved relatere instanser af en bygning til specifikke klasser i disse klassifikationssystemer. Generelt kan man sige at egenskaber og relationer i linked data er mere fleksibelt og semantisk end almindelige klassifikationssystemer og data kan klassificeres med flere klasser på en gang.

### Informationscontainer til dataoverdragelse

Typisk begrænser ARC sig til at kunne tjekke på data der ligger i en IFC-fil, men der er delte meninger om hvorvidt det er hensigtsmæssigt at skulle have alle data i BIM-modellen. I hvert fald er der en begrænsning i de monolitiske BIM-implementeringer vi ser i dag.

ISO 21597-1:2020 Informationscontainer til dataoverdragelse definerer ICDD, som er en standard for udveksling og lagring af filer af forskelligartede typer. ICDD er en standard for opbygningen af containere, som kan indeholde flere modeller, og med RDF er det muligt at beskrive sammenhænge mellem informationer i disse modeller, og det er tilmed muligt at beskrive relationer til ikke-BIM data (billeder, punktskyer, tekstdokumenter, regneark) indeholdt i eller uden for containeren (på nettet). Det er tanken at RDF og linked data-principperne skal lette adgangen til oplysninger i byggeriet. Container-systemet hjælper med at strukturere og sammenkæde heterogene data, så containermodeller er relevante i og på linje med fælles datamiljøer Common Data Environments (CDEs), som letter et centraliseret miljø til forvaltning af både information og tjenester. Således





**Fig. 8.** Hieraki af foldere og filer i en ICDD-container.

sikrer ICDD i kombination med et CDE tilgængelighed af relevante oplysninger på det nødvendige tidspunkt i et byggeprojekt for at lette beslutningstagningen i projektets livscyklus og øge pålideligheden af projektmålene [35].

En .icdd-fil er en zip-fil, som har et standardiseret hieraki af foldere og filer som illustreret i Fig. 8.

## 5 Linked data-understøttet ARC

I et studie fra 2015 undersøger Pauwels og Zhang [31] tre overordnede tilgange til hvad de kalder semantisk regeltjek. Heri forstås at de udføre tjekket på semantiske datamodeller beskrevet som en datamodel baseret på standarderne bag det semantiske web. De beskriver tre overordnede tilgange til regeltjekket:

### 1. Hard-coded rule checking after querying for information:

Den første tilgang er den praksis vi kender til i dag, og som benyttes på BR18-sporet i BART. Reglerne kan beskrives i RDF og OWL, men også i proprietære formater som det der benyttes i SMC. Eksekveringen af reglen afhænger af den software der forstår hvordan den skal afkodes, og resultatets succes er derfor i høj grad afhængigt af den enkelte softwares fortolkning. Desuden er det en udfordring at der ikke ligger nogen standard bag opbygningen af reglen, hvilket betyder at der kun er ringe mulighed for tilretning, og slutteligt påpeges det at metoden begrænser sig til information som er tilgængelig i modellen.