Automated Rule Checking: Hvad er state of the art, og hvordan kan det oversættes til dansk kontekst?

M.H. Rasmussen
mhra@niras.dk og A. Schlachter
ales@niras.dk

NIRAS, Allerød, Danmark

Resumé. Dette studie er udført ifm. metodesporet for projektet Byggeriets Automatiske RegelTjek (BART) og har til formål at kommunikere State of the Art inden for feltet Automated Rule Checking (ARC) til IKT-aktører i den danske byggebranche. Dokumentet indeholder tekniske detaljer og fagspecifikke udtyk, men der er gjort bestræbelser på at formidle de mest komplekse dele af teknologierne på et sprog der er forståeligt for læseren, som ikke på forhånd har kendskab til feltet. Til den mindre tekniske læser findes der et executive summary, som kommunikerer de vigtigste opdagelser til den ikke-tekniske del af målgruppen for BART. Første fase af studiet udmunder i et "point-of-departure", og der har særligt været lagt vægt på at underbygge hidtidige antalgelser om hvad teknologien kan, samt at etablere et en referenceramme og et vokabular for projektet som helhed. I de efterfølgende faser af metodesporet vil der blive fokuseret på hvordan teknologien kan omsættes til en dansk kontekst.

Keywords: Automated Rule Checking (ARC) · Byggeriets Automatiske RegelTjek (BART) · Building Information Modeling (BIM) · Linked Building Data (LBD)

Indholdsfortegnelse

1 Læseguide 3
2 Introduktion 3
3 Nylige erfaringer inden for feltet 6
4 Åbne standarder 13
5 Linked data-understøttet ARC 18
6 Diskussion og perspektivering 20
A Teknologier 24
B Kommercielle cases 26

Forkortelser 29

1 Læseguide

Dette notat er opbygget sådan at metoderne og teknologierne bag ARC først introduceres på et mere overordnet plan. Herefter kommer nogle mere dybdegående afsnit, som dykker ned i den historiske udvikling, den seneste udvikling, samt de grundlæggende potentialer og udfordringer, der ligger i teknologien. I notatet gennemgås også en række etablerede åbne standarder, og det undersøges hvordan disse kan anvendes og særligt kombineres i et ARC-system. Hvert afsnit afsluttes med en opsummering af delkonklusioner, og i det sidste afsnit kommer hovetkonklsionen med forbindelse til BART-projektet som fører til et roadmap-forslag, som de næste faser af metodesporet vil behandle.

Det er ikke notatets formål at give en dybdegående introduktion til teknologierne, men som bilag forefindes nogle seperate notater som giver en grundlæggende introduktion med referencer til hvor mere viden kan opsøges.

Til slut i notatet findes en oversigt over forkortelser og referencer.

Fase 1 Notatet i sin nuværende form er et produkt af første fase af BART-projektets metodespor. Denne fase er undersøgende, og formålet er, at den skal udmunde i et State of the Art, som kan danne grundlag for, hvor fokus skal lægges i de følgende to faser. I de næste faser vil notatet blive udbygget med en metodik for, hvordan teknologierne vil kunne applikeres i en dansk kontekst med software som allerede eksisterer, eller som må forventes at blive tilgængelig inden for nær fremtid. Som udgangspunkt for dette notat har arbejdsgruppen under BART-projektet diskuteret en række hypoteser for hvilke særlige potentialer der ligger i teknologien. Hypoteserne fremhæver kendte udfordringer inden for ARC og skaber retningslinjer for løsningsrummet, som studiet særligt fokuseres imod i dets søgen på relevante løsninger. Følgende hypoteser blev udvalgt som særligt relevante:

Hypoteser Fase 1

- Erfaring viser at ARC stiller store krav til korrekt navngivning af objekter og egenskaber i modellen, og det ville være fordelagtigt at kunne bruge teknologi til automatisk mætning af implicit viden (se Fig. 1), sådan at dette krav bliver mindre restriktivt.
- ARC omfatter imidlertid kun tjek af informationer indeholdt i en Industry Foundation Classes (IFC)-fil, og det ville være en fordel at kunne inddrage flere datasæt (regneark, beskrivelser mv.), som ikke er indeholdt i bygningsmodellen.
- Hovedsporet af BART omhandler ARC ift. krav i bygningsreglementet, men det ville være fordelagtigt med en overordnet metodik for hvordan man beskriver regler sådan at disse også kan stilles på organisations- eller projektniveau. Sådanne regler kunne omfatte funktionskrav, energikrav og krav til driftsmæssige forhold

Hypoteserne behandles systematisk i dette notat og understøttes af den seneste forskning og udvikling.

2 Introduktion

ARC handler, som navnet antyder, om ved brug af software at kunne automatisere hele eller dele af den proces, som er forbundet med at granske projektdokumentationen på et byggeprojekt, for at undersøge om det overholder en række prædefinerede krav. Krav, som hvis disse overtrædes vil medføre afledte negative konsekvenser for det færdige byggeris succesfulde opførelse samt videre drift. Der findes i litteraturen flere synonymer for ARC. Ofte bruges benævnelser som 'requrement checking', 'model checking', 'semantic rule checking', 'regulatory compliance audit', 'model validation' og følgeligt BIM-based Model Checking (BMC) [9, 15, 49, 31].

Anvendelsesområder

I dette studie lægges der særlig vægt på de teknologier og metoder, der helt eller delvist kan automatisere myndighedskrav som de, der er beskrevt i bygningsreglementet, erfaringer fra byggeriets praksis, omfattet af erfaringsblade som BYG-ERFA samt projektspecifikke krav som bygherre

Fig. 1. Når der i studiet refereres til at 'mætte' modellen eller datasættet, så refereres der til den proces, at kombinere det der eksplicit er givet med hvad vi ved, for at kunne udlede det implicitte. Dette er muligt med den gren af AI, som kaldes DL og blandt andet indgår i det semantiske web.

måtte have til det enkelte byggeri. Anvendelsesområderne for ARC er mange, og af Solihin og Eastman (2015) [36] bliver anvendelsesområderne overordnet inddelt i følgende kategorier:

1. Kontrol af hvorvidt en bygningsmodel er korrekt opbygget i overensstemmelse med det sæt standarder eller forud aftalte sæt af krævede betingelser (prægranskning)
2. Kontrol af gældende byggelovgivning
3. Overholdelse af specifikke kundekrav
4. Validering af bygbarhed og andre entreprenørkrav
5. Sikkerhedsregler og andre regler med mulige programmerede korrigerende handlinger, der kan understøtte beslutninger
6. Garantigodkendelser
7. Building Information Modelling (BIM)-datas fuldstændighed med henblik på overdragelse til Facility Management (FM)

Steps for opbygning af et ARC-system

Der er flere trin, der skal overvejes for at udføre ARC, som nævnt i flere artikler [36, 5]. Disse trin giver en generel vejledning i, hvordan et ARC-system kan opbygges, og hvad der er vigtigt at overveje.

1. Konvertering af regler, der er skrevet med henblik på manuel fortolkning, til en form, der kan implementeres på computer; dette er en vigtig udfordring for ældre systemer som fx. bygningsreglementer; i sidste ende kan regelsprog defineres i en form, der er relativt semantisk entydige for mennesker såvel som computere.
2. Beskrivelse af den modelvisning (den relevante delmængde af data), i form af de objekter, attributter og relationer, der er nødvendige, for at understøtte kontrollen. Enten for direkte at kontrollere datasættet, eller for at give kontrolprocedurerne mulighed for at udlede de data, der er nødvendige for kontrollen.
3. Udførelse af reglen, dvs. behandling af logikken i reglen, i form af betingelser, kombinatoriske betingelser og de logiske spørgsmål
4. Rapportering af resultaterne af regelkontrollen til de berørte parter på en sådan måde, at det bliver lettere at korrigere de identificerede problemer.
5. Etablering af procedurer for automatisk korrigering af fejl i reglerne.

I de tidliger faser af BART har der været arbejdet med to stadier af ARC: Prægranskningen som sikrer at de nødvendige oplysninger er tilgængelige (Findes der rum i modellen? Har alle rum en klassifikationskode eller en navngivning der kan oversættes til en? Har alle rum et areal?) og den egentlige granskning som undersøger at diverse regler er overholdt (Er arealbehovene i rumprogrammet opfyldt? Er adgangsforholdene i orden?). Prægranskningen svarer til trin 2 i ovenstående oversigt og den egentlige granskning til trin 3 (Fig. 2).

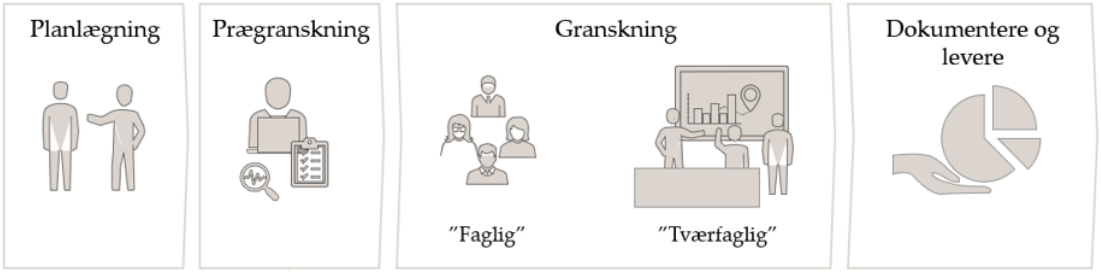

Fig. 2. BART overordnet procesdiagram.

I det næste afsnit fremhæves omfanget af prægranskning seperat, idet det i BART-kontekst anses for et vigtigt forudgående trin for det reelle regeltjek. Dette til trods for at der i de fleste undersøgelser ikke skelnes mellem de to former for granskning og ikke gives en præcis definition af prægranskning.

Prægranskning

Prægranskning er traditionelt den mest grundlæggende ARC-metode og kan udover BIM-modeller benyttes til mange forskellige datasæt (fx. dokumenter, driftskontrakter, osv.). En standardmetode til kontrol af et eller flere datasæt imod en regel er buildingSMARTs Model View Definition (MVD). MVD, der har tætte relationer til IFC, repræsenterer den relevante delmængde af data og gør det muligt at define de nøjagtige krav til regelkontrol, som de eksisterende data opfylder. MVDer anvender regler til at kontrollere datasættets overordnede overensstemmelse og kvalitet, og konsistenstjekket kan sikre, at bygningsobjekter, deres attributter og indbyrdes relationer er i overensstemmelse med et sæt krav eller regler, der er nødvendige for en ARC eller for at opfylde visse (afleverings-) standarder. Senere studie af Zhang (2019) har fundet at MVD er mindre hensigtsmæssig og fleksibel [47], og derfor foreslås det at bruge linked data teknologier i stedet. Formålet er dog det samme. Dette muliggør desuden 'mætning' af datamodellen, som opkvalificerer indholdet i denne (se næste afsnit samt Fig. 1).

Lee et al. (2016) [24] definerer anvendelsesområderne for prægranskning som følger:

1. Kontrol af, at værdien er korrekt
2. Kontrol af værdiens eksistens
3. Kontrol af værdiernes unikhed
4. Kontrol af at typer og klasser er korrekt

I mange kommercielle værktøjer har man brug for at en bruger åbner et stykke software, indlæser datasættet, anvender de nødvendige regler, kører ARC-processen og til sidst modtager en ARC-rapport. De fleste løsninger tillader dog kun visse filformater, og generelt kræves der noget oplæring af brugeren . Især i forbindelse med prægranskning kan denne indsats i form af tid og ressourcer undgås ved at anvende smartere løsninger, som direkte kontrollerer modellens kvalitet og opkvalificerer dens indhold forud for at filen sendes til kunden. Desuden kan denne helt afvises hvis ikke kvaliteten er i orden. Prægranskning er således et værdifuldt værktøj til at kontrollere, at de modtagne data er i overensstemmelse med eksterne og interne regler. Ved at automatisere denne ARC-metode kan indsatsen på klientsiden minimeres betydeligt, og det sikres, at dataene er i overensstemmelse med det krævede format.

Mætning

I de kommercielle værktøjer, som er tilgængelige på markedet, stilles der store krav til kvaliteten af den model, der udføres regeltjek på. Baren er altså højt sat i forhold til den kvalitet der i praksis opleves af de digitale modeller. Det er nødvendigt at bruge den helt rette navngivning af egenskaber ved modelleringen, og det skal ske med netop det klassifikationssystem for egenskaber, som regeltjekket udføres i. Det har imidlertid vist sig at være ufleksibelt. Derfor er der et potentiale i at udnytte teknologier, som gør os i stand til at udlede implicit viden ud fra det, der er eksplicit

beskrevet, for dermed at 'mætte' modellen (Fig. 1). For eksempel er en **IfcDoor** (IFC), en **QQC** (CCS/CCI) og en **bk.dør** (Forvaltnings Klassifikation) det samme, og hvis komponenten sidder i en ydervæg, så er den også en **311** (BIM7AA). WikiPedia beskriver også en dør, og med wikidata kender vi denne som en Q36794. Sidstnævnte er beskrevet med linked data, så den indeholder også referencer som blandt andet fortæller os at døre findes i biler og huse, og så indeholder den i skrivende stund oversættelser på 146 sprog (Tjek selv på `https://www.wikidata.org/wiki/Q36794`).

3 Nylige erfaringer inden for feltet

Potentialet

En udbredt reference ift. besparelsespotentialet ved digitaliseringsdrevet effektivisering i byggebranchen er et studie som amerikanske National Institute of Standards and Technology (NIST) udførte i 2004 [16]. Dette indikerer at problemer med ringe interoperabilitet årligt koster 15,8 milliarder dollars for den amerikanske byggebranche alene. Lignende tendens bekræftes i et mere nyligt studie af McKinsey [3], hvor det anslås at der globalt vil kunne opnås en øget værdiskabelse svarende til 1,7 billioner dollars ved at øge produktiviteten i branchen. En undersøgelse af 139 australske byggeprojekter fra 2011 konkluderede endvidere at 14,2 pct [25]. af de totale projektomkostninger skyldes projekteringsfejl. Ifølge Danmarks statistik var den totale omsætning i den danske byggebranche i 2020 på 311 mia kr [1], så hvis lignende tendenser gør sig gældende her, er det over 40 mia kr der årligt tabes på projekteringsfejl.

ARC er en teknologi, som forventes at kunne nedbringe disse fejl og udløse nogle af de endnu uforløste fordele ved at strukturere information i BIM-modeller [10]. Som buildingSMART skriver i en teknisk rapport fra 2019 [12], så har det typisk været designerne og entreprenørerne, som skulle fremstille argumenter for hvorfor bygherren bør investere i BIM. Argumenter som at det er godt for projektets bundlinje fordi der opnås en højere grad af pålidelighed og effektivitet i design- og anlægsfasen. Med ARC argumenterers det i rapporten for, at bygherrerne (især store developere) direkte kan opnå betydelige fordele ved at sikre overholdelse af reglementer, da det vil reducere sagsbehandlingstiden og reducere fejl og afledte mangeludbedringer. Desuden peges der på de afledte fordele ved at bygningsmodellen kan bruges som omdrejningspunkt for en 'smart building' eller 'digital twin', der kan bruges i driftsfaserne.

En kendt implementering

Den mest omfattende kendte implementering af ARC er Singapores Construction Real Estate Network (CORENET) projekt. I et metastudie af Beach et al. (2020) [5] peges der på CORENETs e-Submission system [18] som det eneste eksempel på en myndighed, der har adopteret ARC som en del af den officielle granskningsproces. Dette projekt er dog ikke længere aktivt, hvorfor der altså ifølge forskerne i dag ikke findes noget eksempel på ARC i funktion.

CORENET stammer helt tillbage til 1993, og rammerne for den del af projektet som omhandler ARC blev lagt i 1999. Singapores regering stillede herigennem en platform til rådighed, som skulle give virksomheder i byggebranchen en række fordele. Platfomen indeholdt 1) "One stop submission" af byggeplaner til godkendelse (**OSSC**), 2) vurdering af bygbarhed (**eBDAS**) samt 3) tjek af integrerede planer for bygning samt tekniske installationer (**IBS/ IBP**). Motivationen for en automatiseret godkendelsesproces (fig. 3) kom fra den traditionelt komplekse proces med indsendelse af ansøgning, kontrol af planlægningskrav, evaluering af forslaget, genindsendelse af forslaget for manglende overensstemmelse og udstedelse af godkendelse, da den involverer mange parter og anvender forskellige kommunikationskanaler [18].

OSSC er et elektronisk indsendelsessystem, der gør det muligt at indsende alle de nødvendige filer til bygningsgodkendelsesprocessen.

eBDAS understøtter parterne i godkendelsesprocessen med et elektronisk værktøj, der kan hjælpe med at udarbejde, måle, konsolidere, rapportere, verificere og analysere byggbarhedsvurderinger. I forhold til granskning giver dette allerede mulighed for at kontrollere, at byggeplanerne er i overensstemmelse med adfærdskodeksen for bygbart design.

[1] https://www.dst.dk/da/Statistik/nyt/NytHtml?cid=32012

IBS/ IBP er den del af løsningen, der mest vedrører ARC fordi det er et kontrolsystem med kunstig intelligens, der automatisk kontrollerer digitale planer for at sikre, at de overholder de lovmæssige krav. Systemet indeholder en konsekvent måde at fortolke lovregler og forordninger på og det sikrer at reglerne for ARC er defineret korrekt.

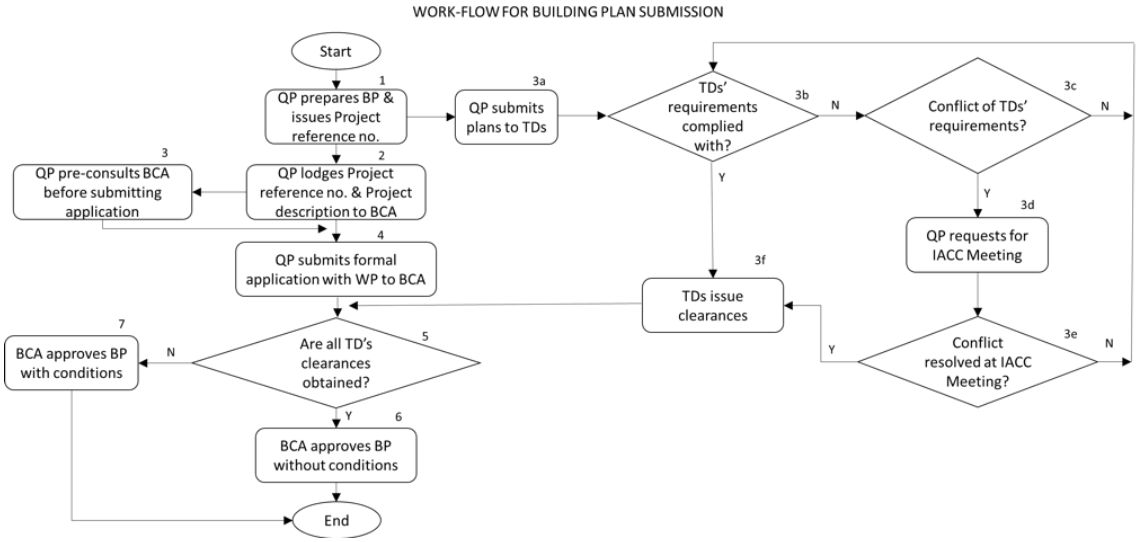

Fig. 3. Work flow i forbindelse med indsendelse af byggeplan i CORENET. **QP** Qualified Person — **BP** Building Plan — **BCA** Building and Construction Authority (Singapore) — **WP** Written Permission — **IACC** Inter-Agency Coordinating Committee — **TD** Technical Department (https://www1.bca.gov.sg/regulatory-info/building-control/building-plan-submission)

Konklusionen af en undersøgelse [18] viser, at størstedelen af de adspurgte virksomheder i Singapore har oplevet fordele ved det automatiserede godkendelsessystem. Undersøgelsesresultatet fremhæver også betydningen af procesomlægning, dvs. at man udfordrer generelt accepteret praksis og etablerede strategier for at skabe procesinnovation og -forbedring.

I tilfælde af CORENET viser det vigtigheden af en fælles process mellem de parter, som skal have deres bygningsplaner tjekket og parter som stiller krav til planen. Denne proces skal resultere i en klar og anvendelig definition af regler og forordninger ift. lov, bygbarhed osv. som kan fortolkes entydigt af et kontrolværktøj. Målsætningen fokuserer således mere på processen end på teknologien, og teknologien skal understøtte den definerede proces.

En brancheundersøgelse

I et nyligt studie fra 2020 af Beach et al. [5] påpeges det at der indtil videre ikke har været nogen meningsfuld adoptering af ARC-teknologi til trods for at BIM-modeller bliver stadig mere modne. Studiet tager udgangspunkt i personer med hovedsageligt BIM-lederroller, og undersøger deres synspunkter på ARC-området. Undersøgelsen viser, at der findes gode løsninger inden for ARC-området, men også mange isolerede tiltag. Selv med de nuværende åbne standarder er der tale om isolerede tiltag, som løser en del af problemet, og der mangler integration i en større sammenhæng.

State of the art består i forfatternes optik af begrænset adoption af granskningsprocesser fra softwareproducenter i kombination med spredte udviklinger af ad-hoc løsninger som ikke skalerer. Det kan derfor ikke umiddelbart videreføres fra et projekt til et andet, og tilgængeligheden for ikke-ekspertbrugere er meget begrænset.

På grundlag af undersøgelsen konkluderer forfatterne at der er hindringer, der står i vejen for en rigtig adoptering af ARC og Tabel 1 viser de forhindringer med en score på over 3. Hindringerne er vurderet på en skala fra 1-4 for, hvor ønskværdig en løsning på denne hindring er (1 ikke påkrævet, 2 ønskelig, 3 meget ønskelig and 4 essentiel).

Manglen på fælles åbne standarder betyder, at der mangler et fælles sprog til at sikre en ensartet fortolkning af regler. Dette er en nødvendighed for at kunne understøtte digitalisering,

Kompetence	Score
Mangel på fælles åbne standarder for regulbestemmelser	3.85
Ingen eksisterende værktøjer, der understøtter et fuldkomment regeltjek før projektmaterialet formelt indsendes	3.46
Ikke muligt at håndhæve krav fra byggeprogram samt lovgivningsmæssige krav kontraktmæssigt	3.45
Mangel på eksisterende regelprocesser til at spore beslutninger og usikkerhed	3.36
Manglende strengt defineret juridisk ansvar for overholdelse	3.33
Manglende kravstillelse ift. aflevering af 'as proposed/designed' og 'as-built' modeller til drift på ikke-offentlige projekter	3.26
Dokumentation og tegningsmateriale har fortsat forrang ift. struktureret asset information (fx i BIM) ved aflevering af hovedprojekt	3.21

Tabel 1. Hindringer for ARC-adoption i UK.

og det er en væsentlig barriere. Desuden fremhæver undersøgelsen, at der mangler løsninger for prægranskning (se afsnit 2), for at sikre at de indsendte dokumenter er parate til at gennemgå granskningsprocessen. En sådan prægranskning ville give kontrolinstansen mulighed for automatisk at afvise ikke-kompatible dokumenter eller filer før det lander på medarbejderens bord. Dermed reduceres mængden af spildtid på opgaver som flytter fokus uden at give fremdrift.

Da mange lovregler ikke er skrevet i et maskinlæsbart sprog er reglerne ikke knyttet til eksplicitte fakta for bygningsdesign eller processen heromkring. Som følge heraf, er det i den nuværende praksis ikke muligt at stille en kontraktmæssig konsekvens for overholdelse af reglerne. Ligeledes kræver bygningsprocessen også en fleksibel beskrivelse af reglerne ift. planlægningsfremdrift hvor reglerne tilpasses efter planlægningssniveauet.

Der er desuden andre forskningsundersøgelser, som giver endnu flere udfordringer med hensyn til ARC's state of the art. Gade (2020) [15] beskriver eksempelvis, at mens BMC-systemer har potentialet til at forbedre bygningernes design, findes der flere tekniske og sociokulturelle udfordringer der skal løses før, at de kan integreres i praksis. Netop de menneskelige faktorer ligger uden for fokusområdet for dette studie, men i senere faser, hvor konkrete værktøjer er til rådighed, vil det være relevant at gå længere ned ad dette spor.

I et teknisk notat udgivet af buildingSMART [12] beskrives det videre at ARC ikke længere blot er et teknisk værktøj hvor man sætter krydser. I stedet for at være et top-down expert system er det begyndt at tage former som er mere semantisk baserede og er baserede på buttom-up analyser og Machine Learning (ML). Det betyder at man skal overveje flere interesseområder eller domæner af ARC og finde løsninger med henblik på større funktionel fleksibilitet og semantisk forståelse.

Det vigtigste resultat af arbejdet i studiet fra Beach et al. [5] er et **roadmap** og en metodisk liste over de næste skridt, der skal bringe byggebranchen tættere på den automatiserede granskingsprocess. Forfatteren opdeler roadmappet i fire faser: *forskning*, *udvikling af proof of concept*, *industrialisering af proof of concept til kommerciel standard* og *skalering af det industrialiserede produkt eller den industrialiserede proces til hele branchen.*

1. **Forskning:** Bestemmelse og prioritering af regler, der kan automatiseres med yderligere inddragelse af politiske beslutningstagere.
2. **Pilotafprøvning:** Udvikling af regelprocesser til at registrere beslutninger, feedback og usikkerhed samt en detaljeret kortlægning af digitaliserede regulerings-/kravs-/standardiseringsprocesser. Derudover udvikling af regler sammen med et fælles sprog og demonstration af arbejdet for at identificere områder, der kan forbedres.
3. **Industrialisering:** Processerne bruges her til at standardisere og digitalisere krav- og produktbeskrivelser for at forhindre variation i specifikationerne og dermen gøre det muligt for kontrolværktøjer at forstå disse beskrivelser og integrere flere dataordbøger og skemaer for at øge konteksten og udbredelsen.
4. **Skalering:** Denne fase indeholder flere kompetencer ift. implementering. Det er af stor betydning at inddrage målgrupper ved at udvikle målgruppespecifik uddannelse og vejledning. De digitaliserede regler skal udnyttes til at foretage en detaljeret analyse af det lovgivningsmæssige landskab og sammenhæng mellem regler og identifikation af overlapninger og huller. Desuden vil gennemførelsen gøre det muligt, at uvikle metoder til automatiseret design af assets

baseret på regler/krav (generativt design). I den forbindelse er det nødvendigt at udvikle eller forbedre konsistente/strukturerede datamodeller og Application Programming Interfaces (APIs) til granskning, for at muliggøre et bredt udbud af ARC systemer.

Når branchen har gennemført dette roadmap, kan følgende vision (fig. 4) om fremtiden for ARC realiseres.

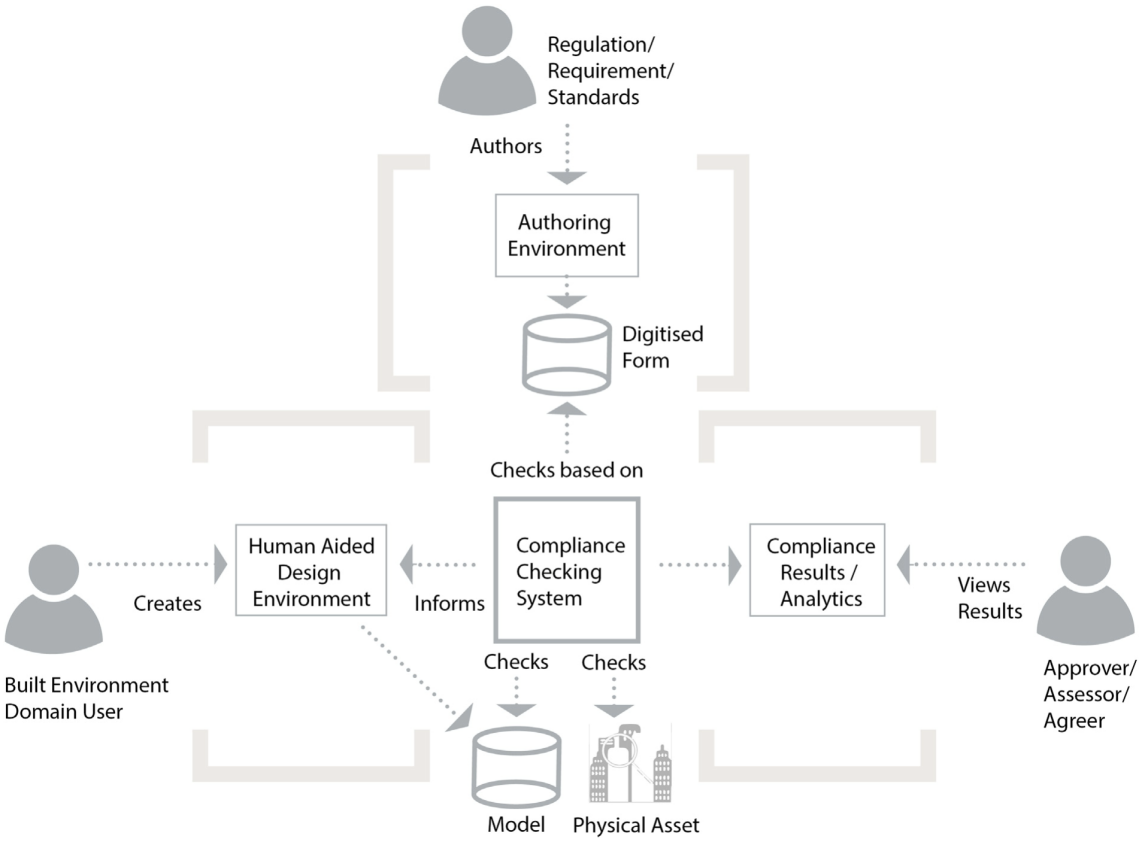

Fig. 4. En vision for ARC

Roadmappet og visionen, som forfatteren præsenterer, giver en lovende og gangbar strategi til at ændre industriens ARC-metode til at bringe industrien til et højere niveau ift. ARC. Forfatterne påpeger endvidere at industrien tror på at ARC er en mulig og ønsket vej såfremt det menneskelige overblik og gennemsigtighed bibeholdes.

Klasser af regler

Definitionen og opsætningen af entydige, maskinlæsbare regler er et væsentligt element for at kunne opnå ARC. I en undersøgelse fra 2015 foreslår Solihin og Eastman (2015) [36] en klassificering af regler til automatiseret udvikling af BIM-regelkontrol. De fire klasser opstiller nogle retningslinjer for den generelle fremgangsmåde for regeludvikling, der omfatter både procesudfordringerne og de teknologiske behov. Undersøgelsen viser, at der i stigende grad er behov for automatisering af kontrollen, hvor veldefinerede regler automatisk kan applikeres med et minimum af brugerindgreb.

Inden de fire klasser præsenteres, fremhæves nogle principper for ARC, som forfatterne har taget hensyn til i deres undersøgelse.

Kravene til regelkontrol omhandler ikke kun eksplicitte objekter og egenskaber, men også mere abstrakte og semantisk rige oplysninger, som ikke altid er til stede i tegninger. Vi har derfor brug for en klar forståelse af de forskellige typer af regler.

Overordnet grupperes reglerne i de syv anvendelsesområder, som blev oplistet i afsnit 2. Som det fremgår, har ARC mange forskellige anvendelsesområder, og ARC-metoder kan ofte anvendes til at kontrollere flere end kun et anvendelsesområde. Artiklen fremhæver, at der opnås mærkbare fordele, hvis ARC hellere kan organiseres *generisk* i forhold til den type af granskning, der kræves, end efter anvendelsesområde. Denne fremgangsmåde kræver følgende overvejelser:

- Generisk organisering af regler *øger genanvendeligheden*, da processen med fortolkning af regler kan udføres én gang og bruges flere gange. Især hvis reglerne er skrevet i et maskinlæsbart format og baseret på en standardterminologi.
- Krav, der ikke er veldefinerede som *bedste praksis*, skal overvejes og formuleres som regler, når de er almindeligt accepteret i branchen (Eksempel: "vægge skal røre pladen nedenunder i geometrisk modellering".)
- Der er en *kløft mellem de data, der skal kontrolleres, og reglernes ordlyd*. For at lukke denne kløft og gøre det muligt at kontrollere specifikke regler, skal implicitte data udledes af de eksplicitte data, (Eksempel: "det frie rum mellem to døre i serie er et forrum".)
- Der er behov for en *standardiseret måde at repræsentere og klassificere bygningsdata på* (IFC). Dette kan ledsages af andre klassifikationssystemer for at gøre ARC mere robust og fleksibelt. Der er desuden behov for at have en standardiseret måde at definere regel-terminologien og relatere det med bygningsdata. Flere oplysninger om åbne standarder og klassifikationer findes i den næste afsnit (afsnit 4). Hvis forskellige terminologier og klassifikationer kan relateres til hinanden, vil det skabe et langt mere effektivt system, der er uafhaengigt af nationale eller domænespecifikke standarder.
- Der findes ingen regel eller klasse af regler, der gælder for hele sættet af bygningsmodeldata og derfor er der behov for at kunne filtrere i datasættet og kun behandle den delmængde, som reglen gælder for.
- ARC skal tage højde for bygningsmodellens udviklingsniveau, eller Level of Development (LOD), så regelkontrol fungerer harmonisk med systemer med forskellige detaljeringsgrader. Regler for fabrikationsspørgsmål skal fx. ikke anvendes på en model med et lavt LOD-niveau, da de nødvendige data ikke findes.

I forhold til disse anbefalinger klassificerer [36] regler for ARC i fire forskellige klasser baseret på typen af kontrolløsning.

Klasse 1 - Regler, der kræver en enkelt eller et lille antal eksplicitte data : Denne klasse af regler kontrollerer eksplicitte data (objekter og egenskaber), der findes i BIM-modellen. Eksempler på klasse 1 regler:

- Er krav til datastrukturoverholdt?
- Findes de nødvendige objekter og egenskaber i henhold til krav fra kunden/bygningskodekset eller regler for regelkontrol.
- Enkle bygningsreglementer (Eksempel: *"bygninger med 5 eller flere etager over jorden skal være forsynet med en eller flere passagerelevatorer"*).
- Data, der kræves i andre klasser af regler (Eksempel *"for at kunne bestemme en lejlighedsenhed, skal alle rum inden for den enhed være medlemmer af en IfcZone, og denne zone skal være navngivet og klassificeret i overensstemmelse med den standardiserede navngivningskonvention eller klassifikation"*)

Klasse 2 - Regler, der kræver enkle afledte egenskabsværdier : Denne klasse af regler skaber nye implicitte værdier ud fra eksplicitte data og reglernes vilkår. Det betyder at implicitte relationer er nødvendige for at opfylde kontrolkravene. Ved at kombinere de grundlæggende BIM-modeldata og disse relationer kan et ARC-værktøj udlede den manglende information som er nødvendig for at kunne kontrollere reglerne. Denne klasse kan indeholde visse aritmetiske eller trigonometriske beregninger fx:

- Anvendelse af tekniske beregninger til at udlede flere objektegenskaber (Eksempelvis *vægt*)
- Finde den lige linjeafstand mellem to punkter (Eksempelvis *bestemmelse af afstand mellem centerpunkt for et vindue og en søjle*)
- Bestemmelse af, om et punkt befinder sig inden for en lukket polygon (Eksempelvis *bestemmelse af radiatorer i et rum*)
- Bestemmelse af, om en geometri befinder sig inden for et lukket volumen

Klasse 3 - Regler, der kræver udvidet datastruktur : Hovedidéen ved denne klasse er at kunne "beregne én gang, bruge flere gange", da sådanne oplysninger typisk kræver omfattende beregninger, der ofte involverer geometriske operationer. I studiet gives eksemplet: *udledning af afstanden mellem detektorer i flade lofter, som ikke må overstige 7 m i gangarealer*. Denne regel kræver en sofistikeret geometrimotor for at skabe et dækningskort ved hjælp af afstandstriangulering mellem detektorer og kan bruges flere gange for forskellige anvendelsesområder.

Klasse 4 - Regler, der kræver bevisførelse Denne klasse af regler kontrollerer og foreslår korrigerende handlinger eller løsninger og fokuserer mere på et bevis for en løsning end på en kontrol af overholdelse. I relation til eksemplet med detektorer vil reglens output være et input, som designeren kan bruge til at korrigere sit design, når der konstateres manglende overensstemmelse med hensyn til den korrekte placering af detektorerne. Dette kunne implementeres direkte i den software, der udarbejder designet, for at give feedback i realtid.

Det konkluderes i artiklen, at det er vigtigt at skelne mellem forskellige klasser af regler og gruppere dem efter typen af kontrolløsning. Regler med lignende kontrolløsninger og kompleksitetsgrad kan udvikles sammen med den samme fremgangsmåde, og man kan drage fordel af skaleringseffekten.

Forfatterne fremhæver, at de første to klasser af regler er relativt enkle at gennemføre, og alligevel har de et enormt potentiale til at bidrage til at forbedre kvaliteten af bygningsmodeller. Især hvis der er en konsistent og nem måde for brugerne at definere reglerne til at kontrollere grundlæggende oplysninger på. Denne fremgangsmåde er tidligere blevet defineret som prægranskning i afsnit 2. De sidste to klasser kræver mere udviklingsindsats og avancerede geometriske operationer. Solibri Model Checker (SMC) er i stand til at bruge klasse 3-regler, men det kræver en stor ekspertise at skabe specifikke og komplekse regler med SMC-udviklere [21]. Der findes ikke nogen løsning, der kan integrere regler med den kompleksitet og de krav, der gælder for klasse 3 og 4.

I det næste afsnit gennemgås en undersøgelse, der anvender åbne standarder til at overvinde de nuværende udfordringer i ARC. I lighed med [36] anvendes forskellige regelkategorier til at klassificere ARCs anvendelsesområde. Zhang (2019) [47] grupperer sine udviklede ARC-funktioner i fire grupper: semantik på skemaniveau, semantik på instansniveau, produktgeometri og rumlige ræsonnementer. Undersøgelsen gennemgås i det følgende afsnit.

En tilgang baseret på åbne standarder

Gennem sit 5 år lange Ph.d.-forløb har Chi Zhang arbejdet med at udvikle metoder til at udføre ARC gennem brug af åbne standarder. I sin afhandling [47] argumenterer han for nogle af de udfordringer der er med de eksisternede kommercielle løsninger (fx SMC og FORNAX). Særligt fremhæver han at regelsæt kun kan beskrives gennem et API som kun er tilgængeligt for ekspertbrugere, at de udviklede regelsæt ikke er transparente for offentlig gennemgang, idet de er afhængige af implementeringen i den enkelte platform, samt at det er omkostningstungt at udvikle nye regler, idet de er baseret på ikke-trivielle programmeringsfærdigheder.

Et forsøg på at øge tilgængeligheden af data i BIM-modeller er BIM Rule Language (BIMRL) [37], som omfatter et simplificeret skema i form af et data-warehouse stjerneskema [17] og en letvægts-geometriengine. Gennem en Extract-Transform-Load (ETL)-proces instatieres sådan et skema som loades i en relationel database, hvorefter det kan tjekkes med en SQL-lignende syntaks med termer fra det definerede dataskema og implementerede funktioner. Selvom der stilles programmerings-interfaces for udvikling af nye funktioner til query-sproget til rådighed, påpeger Zhang dog at ETL-processen er en envejstilgang med et brugerdefineret mapping-program, hvilket begrænser tilgængeligheden og mulighederne for videre brug.

I sin jagt på en skalerbar metode baseret på åbne standarder bygger han to prototyper:

Tilgang A implementeret ved brug af åbne BIM-standarder: IFC til at beskrive bygningsmodellerne, MVD XML (mvdXML) til at repræsentere regelsæt og BIM Collaboration Format (BCF) til at rapportere problemer.

Tilgang B baseret på semantiske webteknologier: bygningsmodellerne stadig beskrevet med IFC, men denne gang i en version baseret på Resource Description Framework (RDF) og regelsæt

repræsenteret ved brug af gængse query- og regelsprog.

Det interessante ved hans rejse er netop at han er nødt til at ændre kurs fra den oprindelige tilgang. Med **Tilgang A**, som bygges oven på det open-source bimserver.org framework møder han nogle udfordringer. Overordnet er metoden ikke tilstrækkeligt ekspressiv, eftersom den kun kan udtrykke et begrænset antal af de testede use-cases fra den hollandske Rgd BIM Standard. Desuden kræves der et dybtgående kendskab til de tekniske detaljer i IFC-datamodellen. Dette resultat er i overensstemmelse med konklusionen af [12], som beskrevet tidligere.

Med **Tilgang B**, der bruger webteknologier, som forudsætter et skema som aldrig er komplet (en såkaldt Open World Assumption (OWA)) giver en større fleksibilitet, og tillader et skalerbart udviklingsmiljø som er fordelagtigt ved udvikling af nye regelsæt og funktionaliteter (se mere om disse teknologier i afsnit 4). For at kunne validere reglerne op imod modellerne er det dog nødvednigt med en hybrid mellem OWA og Closed World Assumption (CWA) - populært sagt er der behov for et sprog til "midlertidigt at lukke verden". Altså tillader man at der løbende vil komme nye objekter i modellerne, men på ethvert givent tidspunkt skal man kunne evaluere hvad der er tilgængeligt i det øjeblik. Denne tilgang evalueres til at give pålidelige resultater. Dog med den begrænsning at use-cases som kræver forbehandling af datasættet ikke er mulige med of-the-shelf semantiske webteknologier. Geometriske analyser som afstandskrav og boolske tests (fx. "rører A og B hinanden¿') kræver egenskaber som først må udledes af IFC bygningsmodellerne, og derefter kræver det analyse af netop disse geometrier. Til dette foreslår Zhang en udvidelse af query-sproget til det semantiske web: SPARQL Protocol and RDF Query Language (SPARQL) som indeholder funktioner til geometriske analyser i 3D. Sådan en udvidelse kendes fra geoSPARQL [4] fra Open Geospatial Consortium (OGC), som dog er begrænset til 2D-geometrier.

Navngivning

Problematikken som løses med BimSPARQL er interessant, men netop geometriske analyser er et af de punkter hvor de eksisterende softwareløsninger er rigtig stærke. Hvor de til gengæld kommer til kort er ift. alsidighed ifm. tjek af egenskaber og relationer mellem objekter. I Zhangs studie peger han på vigtigheden af at have en eksplicit angivelse af disse, og nationale IFC-baserede BIM-standarder som eksempelvis den norske Statsbygg BIM Manual er nødvendige i denne sammenhæng. En sådan standard kan anses som et sæt forretningsregler, som IFC-instanser skal overholde i specifikke sammenhænge.

Der findes nogle eksempler af bestræbelser (fx IFC property sets, Cobuilder og Cuneco Classification System (CCS) - mere om dette i afsnit 4) som forsøger at fastlægge klare termer for simple egenskaber. Disse initiativer giver dog ikke mulighed for at beskrive relationer mellem objekter i modellen. I sproget bag det semantiske web: Web Ontology Language (OWL) skelnes der mellem såkaldte objektegenskaber og datatypeegenskaber. En objektegenskab er en egenskab som forbinder to objekter i en vidensgraf. Eksempelvis kan "har dør"-egenskaben forbinde en objektegenskab på en væg som refererer til en dør, der er placeret i væggen. Datatypeegenskaber er egenskaber som beskriver et objekt, og som har en simpel værdi såsom en tekststreng, en boolsk værdi eller et tal. Eksempelvis kan "dørbredde"-egenskaben beskrive dørens bredde målt i mm. Det er netop objektegenskaber som ikke er omfattet af klassifikationssprog, og det har sine begrænsninger.

Konklusioner

- Der tabes store pengesummer på fejlprojektering, og der er et potentiale i at benytte teknologi til at nedbringe det beløb [16, 3].
- Vi har stadig til gode at se en succesfuld implementering af ARC. Praksis er ringe adoption i software og spredte ad-hoc udviklinger som ikke skalerer og kun i ringe grad kan overføres fra et projekt til det næste [5].
- Der er et behov for automatisering af regeltjekket, hvor veldefinerede regler kan anvendes automatisk med et minimum af brugerindgreb [36].
- 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 projektstadet (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 udfra de oplysninger, der eksplicit er til rådighed [36] (Fig. 1).
- Modeltjek bør integreres med modelskabelsesprocessen for at sikre anvendelighed af regeltjekkene [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 de 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 beregner implicitte data og hvis koblet med en geometri-engine kan man selv integrerer regler af klasse 3 og 4 [47].

4 Åbne standarder

Flere studier i forgående afsnit understregede 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 Ekhom & Häggström [11] defineret som:

> ***Klasse:*** *“et konceptuelt begreb, der henviser til en samling informationsobjekter 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 underindelinger 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, Uniformat, 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.

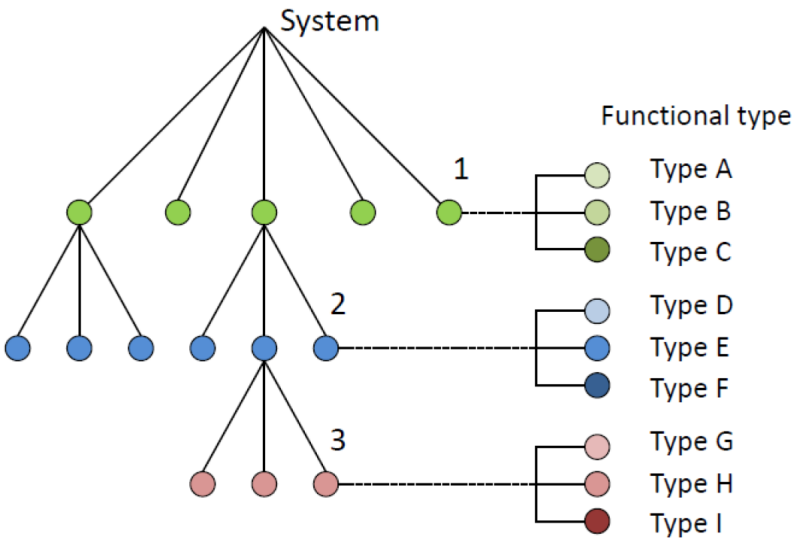

Fig. 5. En klassificeringsstruktur med bestanddele og deres funktionelle typer. Tallene 1-3 angiver niveauer af part-helheds-relationer [11].

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 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)[2] bud på.

Klassifikation af egenskaber

Klassifikation af objekter hører uundgåeligt sammen med specifikation af de egenskaber, som er assosieret 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

[2] https://doc.realestatecore.io/3.2/building.html

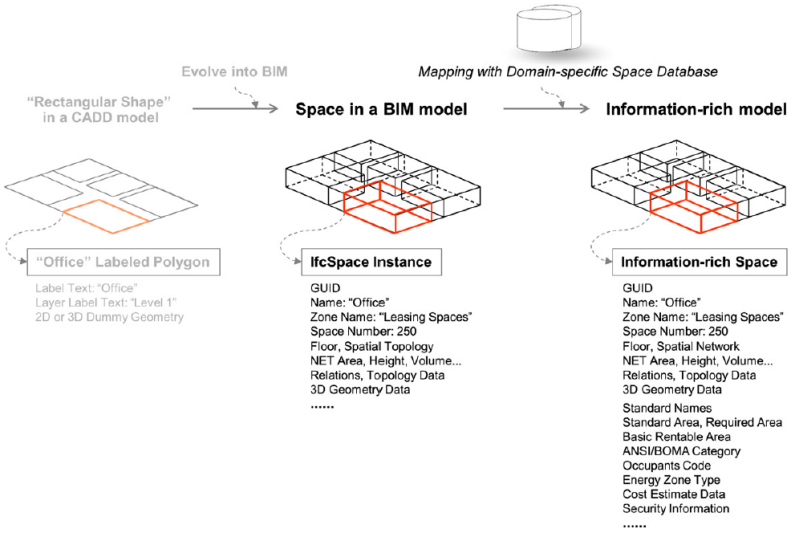

Fig. 6. Oversættelse af simpel CAD-geometri til IfcSpace-instanser til informationsrige rum[22].

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 Organization for Standardization (ISO) (internationalt). Eksempelvis indeholder standarden for beregning af bygningers varmetab (DS 418) indledningsvis en beskrivelse af anvendte begreber og en angivlese 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 beskries 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*[3]. 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*[4]) og brandklasse (*FireRating*[5]). 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 firmaspecifikke aftaler. Med den nye version 5[6] af bSDD som i skrivende stund er under udvikling[7], 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].

[3] https://standards.buildingsmart.org/IFC/RELEASE/IFC2x3/FINAL/HTML/ifcsharedbldgelements/lexical/ifcwall.htm
[4] http://bsdd.buildingsmart.org/#concept/browse/3Yss80qXKHuO00025QrE$V
[5] http://bsdd.buildingsmart.org/#concept/browse/39EXg0qXKHuO00025QrE$V
[6] https://www.buildingsmart.org/users/services/buildingsmart-data-dictionary/
[7] https://github.com/buildingSMART/bSDD/tree/master/2020%20prototype

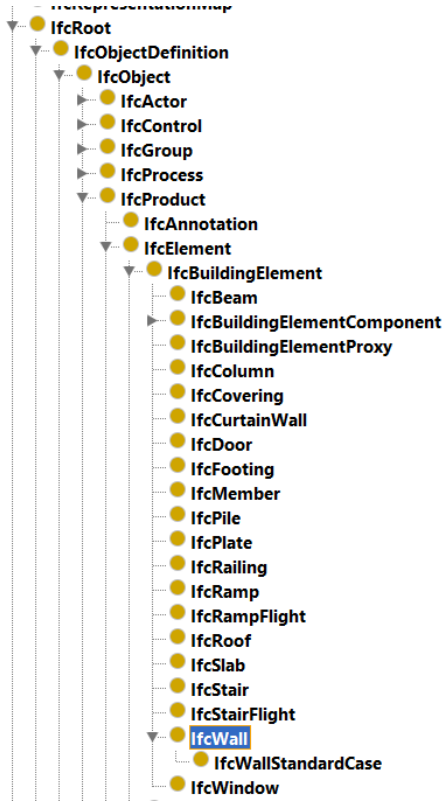

Fig. 7. Udpluk af IFC klasser for bygningselementer visualieseret med Protégé.

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 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

ufleksibelt 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[8], 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, meterologiske, og procesmæssige med hver deres egne standardiserede skemaer, har buildingSMART udgivet en version af IFC som understøtter det semantiske web kaldet ifcOWL.

ifcOWL

IfcOWL er en version af IFC-skemaet, som er beskrevet med webstandarderne RDF[9], RDF Schema (RDFS)[10] og OWL[11]. 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/domæne (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 skeame 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 webaddresser 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

[8] https://forums.buildingsmart.org/t/ifc-modularisation-current-status-and-request-for-feedback/2054
[9] https://www.w3.org/TR/rdf11-primer/
[10] https://www.w3.org/TR/rdf-schema/
[11] https://www.w3.org/TR/owl2-primer/

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 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 modulær [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 modulær 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 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:

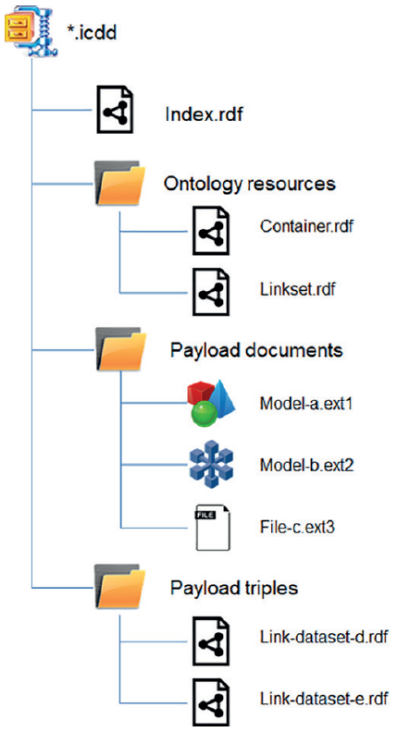

Fig. 8. Hieraki af foldere og filer i en ICDD-container.

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.

2. **Rule-checking by querying**:
   Anden tilgang er at formulere hver regel som en SPARQL query. Dermed er det kun bygningsmodellen, og ikke reglen, som er omfattet af datasættet.

3. **Semantic rule checking with dedicated rule languages**:
   I den tredje tilgang bruges et regel-sprog til at formalisere reglen. Denne fremgangsmåde anses for at være den stærkeste, da der anvendes et standardiseret og semantisk regelsprog, som udelukkende er udviklet med det formål at kontrollere regler, og som kan anvendes til ethvert domæne og formål ift. granskning af datasæt.

En konkret implementering af en tilgang som må anses som en hybrid imellem 2 og 3 er et af Bus et al. (2018/2019) [8, 7]. Her foreslås en systemarkitektur som består i at beskrive de ikke-geometriske informationer med LBD-terminologi (topologi, klasser, egenskaber og fagspecifikke informationer for sig) og de geometri-afledte informationer (centerpunkter, bounding boxes mv.) med kendt terminologi til dette formål. Denne tilgang er ikke langt fra den der benyttes i BimSPARQL [48]. Alle de eksplicitte informationer (abox) gemmes i en triplestore (en grafdatabase til RDF-grafer med en motor til logisk inferens med RDFS og OWL) sammen med regler og queries som fortæller om diverse forhold er opfyldt (se Fig. 9).

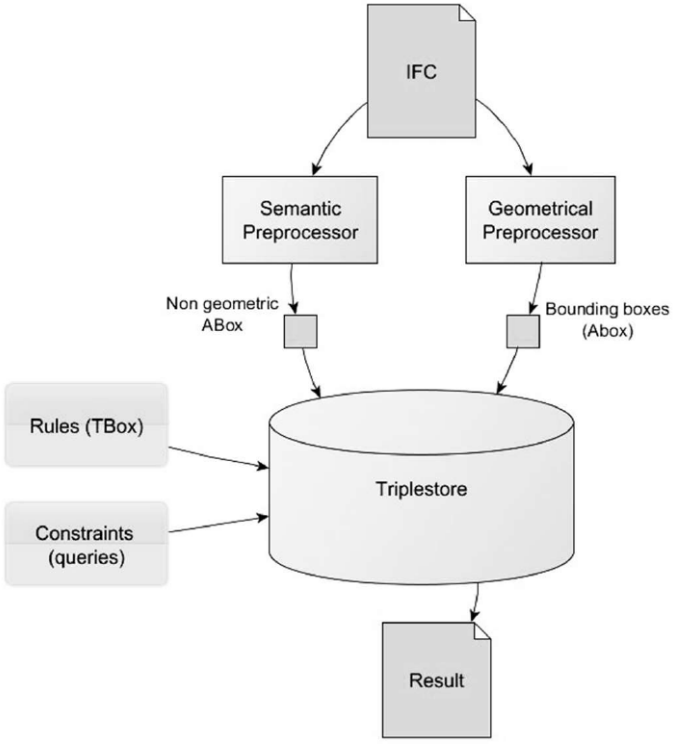

Fig. 9. Arkitektur af den ARC service som foreslås af Bus et al. [7]. **ABox** assertion layer - de eksplicitte informationer — **TBox** terminology layer - ontologierne.

SHACL

Ingen af ovenstående studier benytter web-standarden til beskrivelse af krav til en datamodel, men det må nok tilskrives den årsag at denne er så ny. Shapes Constraint Language (SHACL)[12] er en W3C-anbefaling fra 2017, og definerer grundlæggende et sprog der kan bruges til at validere en RDF graf imod en række krav. En såkaldt SHACL shape beskriver eksplicit hvilke forhold der skal være til stede, og den beskriver ligeledes hvilken besked der skal gives til brugeren (på flere sprog), såfremt den pågældende regel overtrædes samt en gralhed af overtrædelsen (Information, Advarsel eller Overtrædelse). Sproget er kraftigt inspireret af (og kan betragtes som efterfølgeren til) SPARQL Inferencing Notation (SPIN)[13], som er det sprog Zhang benytter i sit studie [47]. Desuden benytter Stolk&McGlinn (2020) [39] det til at validere ifcOWL-modeller (prægranskning) og Oraskari (2021) [28] beskriver det som et sprog der er for LBD hvad mvdXML er for IFC. Derfor er der dele af SHACL, som har været berørt i bygnings- og ARC-relateret kontekst.

6 Diskussion og perspektivering

Opsummering af udfordringer

- Der tabes store pengesummer på fejlprojektering, og der er et potentiale i at benytte teknologi til at nedbringe det beløb [16, 3].
- Vi har stadig til gode at se en succesfuld implementering af ARC. Praksis er ringe adoption i software og spredte ad-hoc udviklinger som ikke skalerer og kun i ringe grad kan overføres fra et projekt til det næste [5].
- Der er et behov for automatisering af regeltjekket, hvor veldefinerede regler kan anvendes automatisk med et minimum af brugerindgreb [36].
- Der er identificeret en mangel på fælles åbne standarder for regelbestemmelser og ARC-håndtering, men SHACL kan potentielt være et element i den rigtige retning [19, 5, 36]

[12] https://www.w3.org/TR/shacl/
[13] https://spinrdf.org/spin-shacl.html

- 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].. SHACL lader dog til at være nyttig til validering af modeldata [39]
- 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 projektstadet (LOD) og det kræver en fleksibel og lettilgængelig tilgang til oprettelse af regler [5, 36].
- Kløften mellem de data, der skal kontrolleres og reglernes ordlyd kræver at ARC-applikationer er i stand til at finde implicitte data udfra de oplysninger, der eksplicit er til rådighed [36] (Fig. 1).
- 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].

Når Beach et al. [5] påpeger at der mangler et fælles sprog til at udtrykke regler, er det ikke helt korrekt. W3C har en anbefaling kaldet SHACL[14], som er et generelt sprog til at opstille randbetingelser for RDF-grafer. Det er muligt at beskrive hvilke egenskaber der forventes på en klasse, hvilken datatype denne egenskab skal have, hvordan den er formateret, hvor mange af dem der må være defineret for én klasse mv. Desuden er det muligt at beskrive et navn og en beskrivelse på flere forskellige sprog, og alvorligheden kan tilmed klassificeres som enten en 'information', en 'advarsel' eller en decideret 'overtrædelse'. I de næste faser vil vi undersøge om SHACL er tilpas ekspressiv til at kunne udtrykke de regler vi ønsker i kontekst af bygninger. Som kort beskrevet i afsnit 5, er der nylige eksempler på at denne teknologi benyttes i kontekst af ARC. Hvis vi skal genbesøge de fem trin for opbygning af ARC-systemer, som blev præsenteret i afsnit 2, så kunne denne med semantiske webteknologier, LBD og SHACL se således ud:

1. Konvertering af regler, der er skrevet med henblik på manuel fortolkning, til en form, der kan implementeres på computer. *Regler kan beskrives i SHACL, men det kræver en oversættelse fra deres eksisterende form i prosatekst, hvilket er en manuel proces som potentielt kan understøttes af Natural Language Processing (NLP).*
2. Beskrivelse af den modelvisning (den relevante delmængde af data), i form af de objekter, attributter og relationer, der er nødvendige, for at understøtte kontrollen. Enten for direkte at kontrollere datasættet, eller for at give kontrolprocedurerne mulighed for at udlede de data, der er nødvendige for kontrollen. *Modeller konverteres til RDF med eksisterende værktøjer*[15] *og forespørgsler på den relevante delmængde af data kan herefter ske med SPARQL . Eventuelt skal der udføres yderligere opkvalificering af modeldata (geometriprocessering, specifikke relationer).*
3. Udførelse af reglen, dvs. behandling af logikken i reglen, i form af betingelser, kombinatoriske betingelser og de logiske spørgsmål. *Enten beskrives SPARQL queries til dette formål, eller hvis reglerne er beskrevet i SHACL processeres denne i et standardværktøj som PySHACL*[16].
4. Rapportering af resultaterne af regelkontrollen til de berørte parter på en sådan måde, at det bliver lettere at korrigere de identificerede problemer. *SHACL returnerer en rapport i standardformat (RDF) som kan formidles grafisk til slutbrugeren.*
5. Etablering af procedurer for automatisk korrigering af fejl i reglerne. *SHACL kan benyttes til konsistenskontrol [39].*

Opsummering af potentialer

- Hvis det foreslåede roadmap følges, får man en metodisk og struktureret fremgangsmåde, der kan bringe byggebranchen tættere på en automatiseret granskingsproces [5].

[14] https://www.w3.org/TR/shacl/
[15] IFCtoLBD: https://github.com/jyrkioraskari/IFCtoLBD
IFCtoRDF: https://github.com/pipauwel/IFCtoRDF
SPARQL generate: https://ci.mines-stetienne.fr/sparql-generate/
RML: https://rml.io/
[16] https://github.com/RDFLib/pySHACL

- Branchen tror på atARC er en mulig og ønsket vej såfremt det menneskelige overblik og gennemsigtighed bibeholdes [5].
- Der kræves en fælles proces mellem regel- eller lovforfattere og de adressater 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.
- Reglerne bør defineres ud fra to perspektiver: anvendelsesområde og afprøvningsmetode [36].
- De første to klasser af regler (regler, der tjekker eksplicitte og enkelte enkle implicitte data), beskrevet af [36] er relativt enkle at gennemføre, og har et enormt potentiale ift. ARC. Især hvis prægranskning kan udføres på en konsistent og lettilgængelig måde.
- 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].
- RDF muliggør brug af flere klassifikationer af objekter/egenskaber, og med RDFS og OWL samt logiske regler giver teknologien mulighed for udledning af implicit viden (Fig. 1) [47, 31].
- SHACL er et standardsprog til at udføre konsistenskontrol af en RDF-datamodel og kan anvendes som et fælles sprog til at definere ARC-regler.

Med udgangspunkt i erfaringerne fra dette studie kan vi definere en række målsætninger for de næste faser af metodesporet.

Roadmap

Som beskrevet af Beach et al. [5] identificerer forskningen fire forskellige procesfaser, som er nødvendige for at gennemføre ARC korrekt i byggebranchen. Disse faser kan sammenfattes som følger.

1. **Forskning og inddragelse af interessenter:** Oplistning og prioritering af regler med henblik på at digitalisere dem.
2. **Pilotafprøvning:** Udvikle regler i et fælles sprog og demonstrere arbejdet for at identificere områder, der kan forbedres.
3. **Industrialisering:** Opbygning af et produkt eller en proces, der opfylder størstedelen af behovene, som afprøves, testes i et repræsentativt miljø, forbedres og forberedes til skalering.
4. **Skalering:** Udvikle en målgruppespecifik uddannelse og vejledning, fastlægge metoder til brugerfeedback og sideløbende udvikle forbedringer.

På baggrund af anbefalingerne fra forskningen er det målet med denne undersøgelse, at omsætte det generelle roadmap til en proceskøreplan, der kan anvendes i praksis, i en BART-kontekst (fig. **??**). Den resulterende proces skal både indeholde de eksisterende leverancer inden for BART, men skal også identificere yderligere initiativer i forbindelse med det foreslåede roadmap for at gøre fremskridt hen imod målet om en korrekt implementering af ARC i byggebranchen. Desuden, som foreslået af Zhang [47], bakker BART-projektet op om den præmis, at ARC-processen baseres på åbne standarder. Det er en nødvendighed for at kunne sikre skalering og national udbredelse.

Fase 1: Tegnologiforståelse af proceskøreplanen i BART-konteksten ligner fase 1 *Forskning og inddragelse af interessenter* fra det generelle roadmap af Beach et al., 2020 [5]. Dette studie er et produkt af første fase, og med dette udgangspunkt er der etableret et velinformeret udgangspunkt for en mulig realisering af next level ARC.

Fase 2: Potentialer og perspektiver indeholder pilotafprøvninger, hvor de identificerede potentialer fra den forgående fase skal afprøves og demostreres gennem mock-ups og proof-of-concept prototyper. Resultaterne af pilotafprøvningerne analyseres og forbedringer identificeres og diskuteres sammen med målgrupperne.

Fase 3: Omsætning til praksis bør ift. Beach et al., 2020 [5] have omhandlet *industrialisering*, men da en egentlig produktudvikling ligger uden for scope af BART-projektet, udelades dette punkt. Næste punkt, skalering, handler også om interessentinddragelse, hvilket er en vigtig del af projektet. Dette punkt vil blive afdækket med udgangspunkt i prototyper snarere end egentlige

produkter. Fasens indhold omfatter desuden uddannelse af interessenterne til at forstå teknologiernes potentiale, og måske tilmed beherske dele de anvendte teknologier. Dette er et skridt i den rigtige retning ift. at kunne tilvejebringe implementeringer der i sidste ende kan bruges af hele branchen.

A Teknologier

RDF

RDF er den datamodel der bruges til at beskrive linked data. RDF kan beskrives i mange formater som XML og JSON, men det mest udbredte kaldes Turtle. I RDF beskrives informationer i en stor "vidensgraf" bestående af knuder der er forbundet med orienterede kanter. To knuder og forbindelsen (kanten) imellem dem kaldes et "statement" eller en "triple", og alt information beskrives i triples.

$$(A) - er\ en \rightarrow (Person)$$
$$(A) - har\ navn \rightarrow ''Peter''$$
$$(A) - kender \rightarrow (B)$$
$$(B) - er\ en \rightarrow (Person)$$
$$(B) - har\ navn \rightarrow ''Mogens''$$

Det er måske svært at se hvordan ovenstående kan give en graf, så derfor er det nedenfor vist i en grafik. De orienterede kanter er vist med pile som indikerer retningen på relationen, og til sammen giver det et lille netværk eller en "vidensgraf".

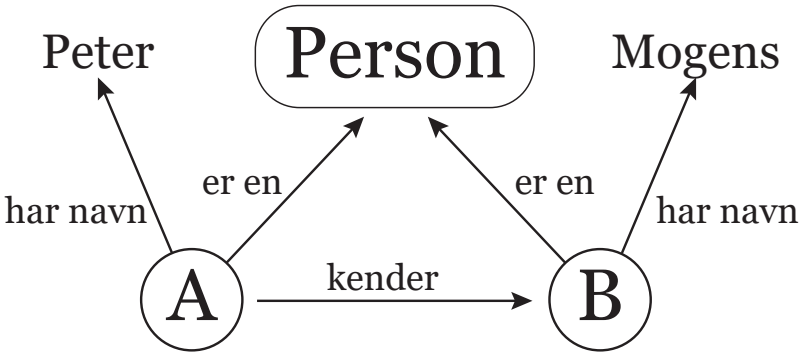

Fig. 10. Vidensgraf.

Ontologier

En ontologi, som af Stunder defineres som en "formel, eksplicit specifikation af en delt konceptualisering" [40] er den del af grafen som beskriver terminologi. I eksemplet fra Fig. 10 drejer det sig de såkaldte objektegenskaber (relationer mellem objekter i grafen): "er en" og "kender", den simple datatype egenskab (tekststrenge, tal, bolske værdier mv.): "har navn" og klassen "Person". Ontologier beskrives også i RDF og med RDFS og OWL er der mulighed for at beskrive logiske relationer imellem de forskellige bestanddele i ontologien (mere om dette i afsnittet omkring logisk inferens). Ifølge "Data on the Web Best Practices" pkt. 15[17] bør det tilstræbes at benytte terminologi som allerede er fastlagt af anden part (helst fra betroede udgiverere som OGC, buildingSMART, ISO osv.), og derfor er det praktisk at en ontologi kan udgives med standard webteknologier og at ontologier kan relatere til termer fra øvrige ontologier.

SPARQL

SPARQL er det query-sprog der benyttes til at udtrække data fra grafen. Det gøres ved at beskrive et mønster i grafen der skal være opfyldt og angive variable for det data der ønskes returneret. For eksempel kan navnene på alle personer udtrækkes med følgende mønster:

[17] https://www.w3.org/TR/dwbp/#ReuseVocabularies

$$?person - er\ en \rightarrow (Person)$$
$$?person - har\ navn \rightarrow ?navn$$

Alt hvad der begyder med ? er en variabel, og konstanter skrives som de er. Altså i dette tilfælde er vi ude efter noget (?person) som er "Person" og for det noget (?person) er vi interesseret i relationen "har navn" med en værdi som vi binder til en anden variabel (?navn). Hvis vi returnerer de bindinger der passer på netop dette mønster får vi resultatet i tabellen herunder.

?person	?name
A	Peter
B	Mogens

Uniform Ressource Identifiers

I regel 1. er det defineret at ting skal navngives med en Uniform Ressource Identifier (URI) og regel 2. angiver at der benyttes http URIer. HTTP URIer er web-adresser. En Uniform Ressource Location (URL) er en mere specifik form for URI og denne angiver placeringen af en fil som for eksempel en hjemmeside. URIer angiver hvad som helst. I det simple eksempel med personen ville det se således ud hvis vi første omgang bytter alle individer ud med http URIer:

$$\text{http://niras.dk/medarbejdere/PHB} - er\ en \rightarrow (Person)$$
$$\text{http://niras.dk/medarbejdere/PHB} - har\ navn \rightarrow ''Peter''$$
$$\text{http://niras.dk/medarbejdere/PHB} - kender \rightarrow \text{http://niras.dk/medarbejdere/MGB}$$
$$\text{http://niras.dk/medarbejdere/MGB} - er\ en \rightarrow (Person)$$
$$\text{http://niras.dk/medarbejdere/MGB} - har\ navn \rightarrow ''Mogens''$$

Vi kan også benytte et prefix hvor vi angiver at nir: er en forkortelse for http://niras.dk/medarbejdere/. Dermed ser eksemplet således ud:

$$\text{nir:PHB} - er\ en \rightarrow (Person)$$
$$\text{nir:PHB} - har\ navn \rightarrow ''Peter''$$
$$\text{nir:PHB} - kender \rightarrow \text{nir:MGB}$$
$$\text{nir:MGB} - er\ en \rightarrow (Person)$$
$$\text{nir:MGB} - har\ navn \rightarrow ''Mogens''$$

Næste step er relationerne og klasserne. Relationen "er en" benyttes til at angive at et individ tilhører en specifik klasse. Dette er indeholdt i RDF og har URIen http://www.w3.org/1999/02/22-rdf-syntax-ns#type. Relationerne "har navn" og "kender" samt klassen "Person" er indeholdt i et kendt skema kaldet friend of a friend (FOAF) (http://xmlns.com/foaf/0.1/). Vi kan derfor udbytte vores egne beskrivelser med globalt anerkendte termer sådan at vores datasæt giver mening for andre der måtte have interesse i at læse det. I denne forbindelse kan vi også fjerne pilene, som blev benyttet i ovenstående eksempler for simpelthen at beskrive grafen i Turtle syntaks:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix nir: <http://niras.dk/medarbejdere/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

nir:PHB rdf:type foaf:Person .
nir:PHB foaf:name "Peter" .
nir:PHB foaf:knows nir:MGB .
nir:MGB rdf:type foaf:Person .
nir:MGB foaf:name "Mogens" .
```

Det at der benyttes http URIer er smart fordi enhver med adgang til en internetbrowser kan slå termen op, for at få mere information om hvad denne betyder. For eksempel beskriver FOAF at ” The knows property relates a Person to another Person that he or she knows (. . .)”.

Logisk inferens

Udover at kunne fastlægge termer og definere deres betydning, så er det også muligt at beskrive regler for inferens. Altså implicit viden, der kan udledes fra hvad der allerede er kendt. Det er muligt med RDFS og OWL.

$$\text{hvad jeg læste} + \text{hvad jeg ved} = \text{hvad jeg infererer}$$

Med RDFS er det eksempelvis muligt at beskrive hieraker af klasser og egenskaber så hvis mountainbikes er en cykel og en cykel er et køretøj, så er mountainbikes også køretøjer.

Med OWL er det muligt at beskrive logik på et højere niveau. Eksempelvis at relationen “kender” er symmetrisk så hvis Peter kender Mogens, så er det også implicit givet at Mogens kender Peter. Det er faktisk lige nøjagtig sådan “foaf:knows” er defineret, og dermed vil en standard OWL regelmotor være i stand til at udlede den implicitte viden der er vist i figuren herunder:

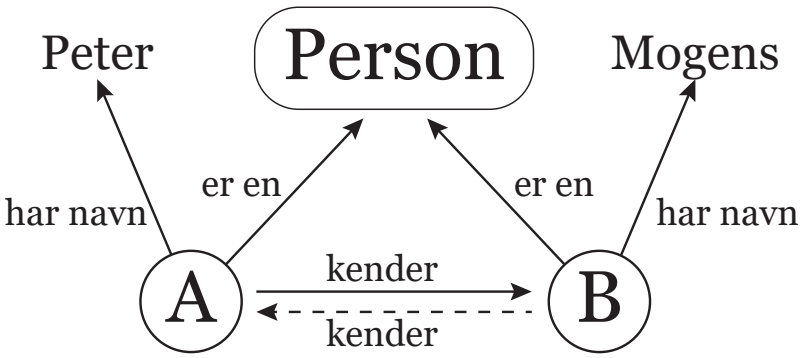

Fig. 11. Vidensgraf.

SHACL

SHACL har siden juli 2017 været en W3C anbefaling hvilket betyder at det er de facto-standarden for at validere RDF-grafer. SHACL giver mulighed for at beskrive regler i såkaldte shapes. De beskrives i RDF og kan derfor gøres tilgængelige online. For eksempel kunne en shape beskrive at bredden på adgangsveje skal være minimum 1,30 m. Dette på en eksplicit måde som bygningsreglementet kunne gøre offentligt tilgængelig på deres hjemmeside.

Apache Jena Fuseki er en gratis triplestore som understøtter SHACL. I et offentligt GitHub repository1 har vi forsøgt at lave en lille test hvor vi tester en BIM-model i RDF format ift. de regler som er defineret af [46]. Det undersøges at der er en byggegrund med minimum én bygning som har præcis ét nulpunkt. Enhver bygning i projektet skal endvidere have mindst én etage som indeholder mindst ét rum.

B Kommercielle cases

I dette bilag fremhæves to af de mest kendte kommercielle ARC-applikationer i byggebranchen, og der afsluttes med et kort resumé af de udfordringer, som ARC står over for.

dRofus

dRofus er et databasesystem til styring af byggeprogrammer, tekniske/funktionelle krav og udstyr fra den tidlige planlægningsfase og gennem hele byggeprojektet. Det omhandler strukturerede rumkrav og rumdataarkiv ved at analysere data fra IFC-modeller (https://www.drofus.com/).

Løsningen fungerer som et lager af behovsrelaterede oplysninger i forbindelse med bygningsmodellen og kan repræsentere krav til rum, udstyr, bygningsservices osv. Igennem "Room Data Sheet" kan man sturturere og gemme rumkrav i forskellige kategorier. Der er dog kun en begrænset grad af automatiseret kontrol af, om modellen rent faktisk overholder de fastsatte regler, i dRofus. I en undersøgelse forsøgte man, at automatisere kontrollen af bygherrekrav til byggeprojekter med forskellige softwareapplikationer og konkluderede, at kun 12% af i alt 138 krav kunne kontrolleres automatisk af dRofus, selv om alle bygherrekrav i undersøgelsen kunne indsættes i rumdatarkiv. En yderligere begrænsning af dRofus er, at det hovedsagelig fokuserer på rumkrav og ikke tager højde for krav på højt niveau, f.eks. navngivningskonventioner for modeldata [2].

I undersøgelsen blev også det kommercielle værktøj SMC afprøvet, som er et mere sofistikeret program til modelkontrol end dRofus. SMC kunne dog også kun automatisere kontrollen af 41% af de givne krav på grund af manglende fleksibilitet og begrænsninger i forhold til, hvilke data der kan kontrolleres. Dette værktøj introduceres i følgende afsnit.

SMC

SMC er en regelbaseret modelkontrolplatform, der læser en IFC-model og kontrollerer dens kvalitet og overensstemmelse på grundlag af foruddefinerede regler. Ud over at kunne identificere sammenstød mellem forskellige fagmodeller (clash detection), giver forskellige regler mulighed for at kontrollere krav til tilgængelighed og frigivelse, revision og kontrol af oplysninger mellem relaterede komponenter (https://www.solibri.com/solibri-office).

Selvom SMC giver mulighed for at definere og modificere regler, mangler det ligesom andre kommercielle værktøjer en betydelig grad af fleksibilitet til at tilpasse regler til nye eller andre regelsæt. Dette betyder, at det er vanskeligt at anvende SMC ud over det anvendelsesområde, der er defineret i de forordninger og andre regelsæt, der oprindeligt er blevet udviklet. SMC indeholder snesevis af regelsæt for forskellige bygningsreglementer. Regelsættene er parametriske og kan ændres for at imødekomme nye krav. Reglerne kan imidlertid ikke oprettes frit, da der ikke er åben adgang til API'et (Application Program Interface), og det er ikke altid muligt at definere de ønskede betingelser. SMC-brugere kan derfor selv kombinere eksisterende regelsæt og konfigurere individuelle regelparametre, men da disse regelsæt er hard-codet ind i softwaren, skal nye regler tilføjes af Solibris egne udviklere, når der er behov for det [27].

SMC stiller desuden store krav til kvaliteten af modellen, og det er nødvendigt at bruge den helt rette navngivning af egenskaber, hvilket skal ske med netop det klassifikationssystem for egenskaber som reglen er skrevet i.

Analysen af de nuværende kommercielle værktøjer til ACR konkluderer, at de nuværende automatiserede kodekontrolmetoder stadig er afhængige af at oversætte oplysninger fra lovkrav til eksplicitte krav og senere til kodificerede krav ved hjælp af logiske regler, hvilket grundlæggende afhænger af menneskelig involvering. Ved at øge den procentdel af kravene, der kan kontrolleres automatisk, vil eksperterne have mere tid til at vurdere opfyldelsen af de krav, der kræver deres viden og en vis grad af subjektivitet [38]

Identificerede mangler ved aktuelle løsninger

Undersøgelsen af [2] viser, at der ikke findes ét kommercielt værktøj, som kan opfylde alle de krav, der stilles i et byggeprojekt. Begrænsningerne ved det undersøgte værktøj er opsummeret i følgende liste.

1. Begrænset grad af automatisering [2]
2. Begrænset mulighed for at tjekke kvalitative regler [2]
3. Ufleksibelt ift. til den krævede datakvalitet og det krævede format [12, 2]
4. Regelkontrolapplikationer skal specialtilpasses for at kunne anvendes til nye regler eller andre regelsæt. Dette kræver en betydelig grad af fleksibilitet, som ikke er tilgængelig i eksisterende applikationer [27].

5. Styring af data-interoperabilitet [45]
6. Dataene/modellen skal oprettes i overensstemmelse med den valgte klassifikation og de semantiske beskrivelser, som danner grundlaget for reglerne. Manglende overholdelse af de semantiske, syntaktiske eller generelle krav til modelleringsreglerne resulterer i en fejlfortolkning af regelkontrolværktøjet. Dette gør det nødvendigt at foretage en forudgående kontrol af, om de semantiske, syntaktiske og generelle modelleringskrav er opfyldt, før yderligere krav kan kontrolleres [23].

Forkortelser

ADK adgangskontrol
AI kunstig intelligens
ARC Automated Rule Checking
API Application Programming Interface
ASHRAE American Society of Heating, Refrigerating and Air-Conditioning Engineers

BART Byggeriets Automatiske RegelTjek
BCF BIM Collaboration Format
BIM Building Information Modelling
BIMRL BIM Rule Language
BMC BIM-based Model Checking
BOT the Building Topology Ontology
bSDD buildingSMART Data Dictionary

CAD Computer Aided Design
CCI Cuneco Classification System International
CCS Cuneco Classification System
CDE Common Data Environment
CEN European Committee for Standardization
CityGML City Geography Markup Language
CORENET Construction Real Estate Network
CWA Closed World Assumption

DBK dansk byggeklassifikation
DL Description logics
DS Dansk Standard

ETL Extract-Transform-Load
ETSI European Telecommunications Standards Institute

FM Facility Management
FOAF friend of a friend

GIS Geographic Information System
GUID globally unique identifier

ICDD Information Container for linked Document delivery
IFC Industry Foundation Classes
IoT Internet of Things
ISO International Organization for Standardization

JSON JavaScript Object Notation
JSON-LD JSON for Linked Data

LBD Linked Building Data
LOD Level of Development

ML Machine Learning
MVD Model View Definition
mvdXML MVD XML

NLP Natural Language Processing

OGC Open Geospatial Consortium
OWA Open World Assumption

OWL Web Ontology Language

PAS Publicly Available Specifications

REC Real Estate Core
RDF Resource Description Framework
RDFS RDF Schema

SfB Samarbetskommitten for Byggnadsfragor
SHACL Shapes Constraint Language
SMC Solibri Model Checker
SPARQL SPARQL Protocol and RDF Query Language
SPIN SPARQL Inferencing Notation

URI Uniform Ressource Identifier
URL Uniform Ressource Location

W3C World Wide Web Consortium
WKT Well-Known Text

XML extensible markup language

Referencer

1. Afsari, K., Eastman, C.M.: A comparison of construction classification systems used for classifying building product models. In: 52nd ASC Annual International Conference Proceedings. pp. 1–8 (2016)
2. Baldauf, J.P., Formoso, C.T., Tzortzopoulos, P., Miron, L.I.G., Soliman-Junior, J.: Using building information modelling to manage client requirements in social housing projects. Sustainability **12**(7), 2804 (Apr 2020). https://doi.org/10.3390/su12072804, https://doi.org/10.3390/su12072804
3. Barbosa, F., Woetzel, J., Mischke, J.: Reinventing construction: A route of higher productivity. Tech. rep., McKinsey Global Institute (2017)
4. Battle, R., Kolas, D.: Geosparql: enabling a geospatial semantic web. Semantic Web Journal **3**(4), 355–370 (2011)
5. Beach, T.H., Hippolyte, J.L., Rezgui, Y.: Towards the adoption of automated regulatory compliance checking in the built environment. Automation in Construction **118**, 103285 (2020)
6. Beetz, J., Van Leeuwen, J., De Vries, B.: Ifcowl: A case of transforming express schemas into ontologies. Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AI EDAM **23**(1), 89 (2009)
7. Bus, N., Muhammad, F., Fies, B., Roxin, A.: Semantic topological querying for compliance checking. In: eWork and eBusiness in Architecture, Engineering and Construction, pp. 459–464. CRC Press (2018)
8. Bus, N., Roxin, A., Picinbono, G., Fahad, M.: Towards french smart building code: Compliance checking based on semantic rules. arXiv preprint arXiv:1910.00334 (2019)
9. Dimyadi, J., Amor, R.: Automated building code compliance checking–where is it at. Proceedings of CIB WBC **6**, 1 (2013)
10. Eastman, C., min Lee, J., suk Jeong, Y., kook Lee, J.: Automatic rule-based checking of building designs. Automation in Construction **18**(8), 1011–1033 (dec 2009). https://doi.org/10.1016/j.autcon.2009.07.002, https://doi.org/10.1016%2Fj.autcon.2009.07.002
11. Ekholm, A., Häggström, L.: Building classification for bim–reconsidering the framework. In: CIB W78-W102 2011: International Conference. CIB (2011)
12. El-Diraby, T.: Beyond e-permitting : Framing the business case for automated rule checking in aec in the era of big data. Tech. rep., buildingSMART (2019)
13. Esnaola-Gonzalez, I., Bermúdez, J., Fernandez, I., Arnaiz, A.: Semantic prediction assistant approach applied to energy efficiency in tertiary buildings. Semantic Web **9**(6), 735–762 (2018)
14. Fuchs, S., Scherer, R.J.: Multimodels—instant nd-modeling using original data. Automation in Construction **75**, 22–32 (2017)
15. Gade, P.N.: Ecological bim-based model checking (2020)
16. GCR, N.: Cost analysis of inadequate interoperability in the us capital facilities industry. National Institute of Standards and Technology (NIST) pp. 223–253 (2004)
17. Giovinazzo, W.A.: Object-oriented data warehouse design: building a star schema. Prentice Hall PTR (2000)
18. Goh, B.H.: E-government for construction: The case of singapore's corenet project. In: Research and practical issues of enterprise information systems II, pp. 327–336. Springer (2007)
19. Greenwood, D., Lockley, S., Malsane, S., Matthews, J.: Automated compliance checking using building information models. In: The Construction, Building and Real Estate Research Conference of the Royal Institution of Chartered Surveyors, Paris 2nd-3rd September. RICS (2010)
20. Jørgensen, K.A.: Classification of building object types: Misconceptions, challenges and opportunities (2011)
21. Lee, J.M.: Automated checking of building requirements on circulation over a range of design phases. Ph.D. thesis, Georgia Institute of Technology (2010)
22. Lee, J.K., Lee, J., Jeong, Y.s., Sheward, H., Sanguinetti, P., Abdelmohsen, S., Eastman, C.M.: Development of space database for automated building design review systems. Automation in Construction **24**, 203–212 (2012)
23. Lee, Y.C., Eastman, C.M., Lee, J.K.: Validations for ensuring the interoperability of data exchange of a building information model. Automation in Construction **58**, 176–195 (Oct 2015). https://doi.org/10.1016/j.autcon.2015.07.010, https://doi.org/10.1016/j.autcon.2015.07.010
24. Lee, Y.C., Eastman, C.M., Solihin, W., See, R.: Modularized rule-based validation of a BIM model pertaining to model views. Automation in Construction **63**, 1–11 (Mar 2016). https://doi.org/10.1016/j.autcon.2015.11.006, https://doi.org/10.1016/j.autcon.2015.11.006
25. Lopez, R., Love, P.E.: Design error costs in construction projects. Journal of construction engineering and management **138**(5), 585–593 (2012)
26. Ma, L., Sacks, R.: A cloud-based bim platform for information collaboration. In: ISARC 2016-33rd International Symposium on Automation and Robotics in Construction. vol. 33, pp. 581–589. IAARC (The International Association for Automation and Robotics in ... (2016)
27. Martins, J., Rangel, J., Abrantes, V.: Automated rule-checking - a tool for design development. In: 41st IAHS WORLD CONGRESS - Sustainability and Innovation for the Future. vol. 41, pp. 1–8 (2016)

28. Oraskari, J., Senthilvel, M., Beetz, J.: Shacl is for lbd what mvdxml is for ifc. In: CIB W78 Conference (2021)
29. Pauwels, P., Roxin, A.: Simplebim: From full ifcowl graphs to simplified building graphs. In: Ework and ebusiness in architecture, engineering and construction, pp. 11–18. CRC Press (2017)
30. Pauwels, P., Terkaj, W.: Express to owl for construction industry: Towards a recommendable and usable ifcowl ontology. Automation in Construction **63**, 100–133 (2016)
31. Pauwels, P., Zhang, S.: Semantic rule-checking for regulation compliance checking: An overview of strategies and approaches. In: 32rd international CIB W78 conference (2015)
32. Rasmussen, M.H.: Digital Infrastructure and Building Information Modeling in the Design and Planning of Building Services. Ph.D. thesis, Technical University of Denmark, Department of Civil Engineering (2019)
33. Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P.: Bot: the building topology ontology of the w3c linked building data group. Semantic Web (Preprint), 1–19 (2019)
34. Schneider, G.F.: Towards aligning domain ontologies with the building topology ontology. In: Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017) (2017)
35. Senthilvel, M., Oraskari, J., Beetz, J.: Common data environments for the information container for linked document delivery (06 2020)
36. Solihin, W., Eastman, C.: Classification of rules for automated BIM rule checking development. Automation in Construction **53**, 69–82 (may 2015). https://doi.org/10.1016/j.autcon.2015.03.003, https://doi.org/10.1016%2Fj.autcon.2015.03.003
37. Solihin, W., Eastman, C., Lee, Y.C., Yang, D.H.: A simplified relational database schema for transformation of bim data into a query-efficient and spatially enabled database. Automation in Construction **84**, 367–383 (2017)
38. Soliman-Junior, J., Formoso, C.T., Tzortzopoulos, P.: A semantic-based framework for automated rule checking in healthcare construction projects. Canadian Journal of Civil Engineering **47**(2), 202–214 (Feb 2020). https://doi.org/10.1139/cjce-2018-0460, https://doi.org/10.1139/cjce-2018-0460
39. Stolk, S., McGlinn, K.: Validation of ifcowl datasets using shacl
40. Studer, R., Benjamins, V.R., Fensel, D.: Knowledge engineering: Principles and methods. Data & knowledge engineering **25**(1-2), 161–197 (1998)
41. Succar, B., Sher, W., Williams, A.: Measuring bim performance: Five metrics. Architectural Engineering and Design Management **8**(2), 120–142 (2012)
42. Terkaj, W., Pauwels, P.: A method to generate a modular ifcowl ontology. In: 8th International Workshop on Formal Ontologies meet Industry. vol. 2050 (2017)
43. Terkaj, W., Schneider, G.F., Pauwels, P.: Reusing domain ontologies in linked building data: the case of building automation and control. In: 8th International Workshop on Formal Ontologies meet Industry. vol. 2050 (2017)
44. Uhlenbruch, W., Stoy, C., et al.: Investigating information needs for specific processes in real estate to prioritise data. Tech. rep., European Real Estate Society (ERES) (2019)
45. Vass, S., Gustavsson, T.K.: Challenges when implementing BIM for industry change. Construction Management and Economics **35**(10), 597–610 (Apr 2017). https://doi.org/10.1080/01446193.2017.1314519, https://doi.org/10.1080/01446193.2017.1314519
46. Werbrouck, J., Senthilvel, M., Beetz, J., Pauwels, P.: A checking approach for distributed building data. In: 31. Forum Bauinformatik: 11.–13. September 2019 in Berlin. Proceedings. p. 173. Universitätsverlag der TU Berlin (2019)
47. Zhang, C.: Requirement checking in the building industry: enabling modularized and extensible requirement checking systems based on semantic web technologies. Ph.D. thesis, Technische Universiteit Eindhoven (2019)
48. Zhang, C., Beetz, J., de Vries, B.: Bimsparql: Domain-specific functional sparql extensions for querying rdf building data. Semantic Web **9**(6), 829–855 (2018)
49. Zhang, C., Beetz, J., Weise, M.: Interoperable validation for ifc building models using open standards. Journal of Information Technology in Construction (ITcon) **20**(2), 24–39 (2015)