



Executive Summary

Automated Rule Checking (ARC)

En opsummering af den nyeste, internationale forskning inden for automatiske regeltjek i byggeriet.

Juni 2021

**BART-projektet er et samarbejde mellem
9 byg- og driftsherrer:**

Bygningsstyrelsen, Københavns Universitet, Københavns Lufthavne, Københavns Kommune, Danmarks Tekniske Universitet, Aarhus Kommune, 4K-samarbejdet (Gentofte, Gladsaxe, Lyngby-Taarbæk og Rudersdal kommuner), samt vidensorganisationerne buildingSMART og Molio. NIRAS A/S står for projektledelsen og gennemførelsen af projektet.

Denne rapport er udarbejdet af NIRAS A/S:

Mads Holten Rasmussen og Alexander Schlachter og Sidsel Nymark Ernsten.

Forside illustration: NCC

Kontakt: mhra@niras.dk

Publiceret i juni 2021

Automated Rule Checking (ARC)

En opsummering af den nyeste, internationale forskning inden for automatiske regeltjek i byggeriet.

Executive Summary

Denne rapport opsummerer den nyeste, internationale forskning inden for Automatic Rule Checking (ARC) teknologi. Ud fra et litteraturstudie af ca. 50 videnskabelige publikationer, beskriver vi hvad ARC-teknologien er og hvilket potentiale og udfordringer teknologien har. Litteraturstudiet viser, at selvom ARC teknologien har et kæmpe potentiale (ca. 40 mia kr alene i Danmark), er den meget svær at implementere i praksis. ARC-teknologien har brug for et skub i den rigtige retning, og den nyeste forskning peger på, at semantiske webteknologier kan give ARC teknologien det nødvendige skub. Med udgangspunkt i forskningen og erfaringer fra international praksis, beskriver vi, hvordan en kobling af ARC og semantiske webteknologier kan se ud i praksis. Rapporten er udarbejdet som en del af BART-projektet i maj 2021.

Introduktion

Automatic Rule Checking (ARC) benytter software til at undersøge om byggeprojekter overholder en række prædefinerede krav til byggeriet. ARC handler altså om at automatisere hele eller dele af granskningsprocesserne i et byggeprojekt.

Det er anerkendt i litteraturen at der ligger et stort potentiale i at kunne automatisere dele af denne proces. Ikke blot for at øge produktiviteten, men særligt med henblik på at kunne nedbringe antallet af omkostningstunge fejl der bliver begået. Lopez & Love (2011) påpeger at 14,2 pct. af de totale projektkomkostninger i 139 undersøgte australske projekter skyldes projekteringsfejl, og dette tal er endda ret ens for forskellige entreprisereformer og projekttyper. Hvis dette tal kan overføres til en dansk kontekst, hvor byggebranchen i 2020, ifølge Danmarks Statistik, havde en total omsætning på 311 mia kr¹, betyder det, at der bliver spildt omkring 40 mia. kr. på projekteringsfejl om året. Der er altså et stort økonomisk potentiale i at udnytte teknologi til at identificere projekteringsfejl tids nok til at de kan blive udbedret. Ifølge Beach (2020) har vi desværre stadig til gode at se en succesfuld implementering af ARC.

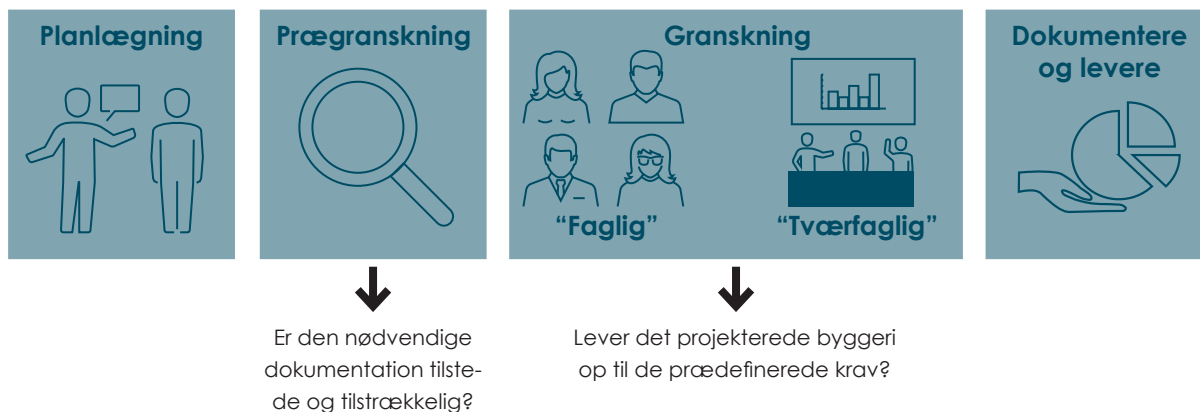
ARC er ikke udbredt i byggebranchen, og er også kun delvist implementeret i tilgængelig software-produkter. Det findes dog forskellige ad-hoc udviklede løsninger som ikke skalerer og kun i ringe grad kan overføres fra et projekt til det næste. Kommercielle værktøjer som Solibri Model Checker (SMC) er i stand til at løse dele af ARC-opgaven, men der er nogle udfordringer med fx:

→ **Prægranskning:** Inden projektet granskes, skal det sikres at de nødvendige oplysninger er tilgængelige og at kvaliteten af oplysningerne er tilstrækkelig. Fx skal det sikres at alle rum i modellen har et areal og en klassifikationskode. Der mangler værktøjer, der kan understøtte denne prægranskningsproces, (se figur 1).

→ **Datakvalitet.** Data i bygningsmodellen er ofte mangelfuld. Uden de rette data i modellen er der ikke noget grundlag at udføre den reelle granskning på. Hvis man for eksempel vil tjekke om døre, der grænser op mod et handicapp toilet, lever op til kravene til dørbredde, er det nødvendigt at oplysninger om dørbredde, rumtype mv. fremgår af modellen. Nogle gange kan disse oplysninger findes ved at opkvalificere nogle af de data, der allerede findes i modellen.

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

→ **Proprietær software.** Reglerne skrives i et proprietært sprog, som kræver indgående kendskab til modellens tekniske opbygning. Pga. disse udfordringer mener forskerne ikke, at ARC-værktøjerne kan skalere i deres nuværende form (Baldauf et al. 2020; . El-Diraby 2019; Zhang 2019; Vass og Gustavsson 2017). Der er brug for en ny tilgang til ARC.



Figur 1: En typisk granskningsproces

4 klasser af regler

ARC har mange forskellige anvendelsesområder som konsistenskontrol, overholdelse af lovgivning, bygherrekrav, bygbarhed mv. og grundlæggende ARC-metoder kan ofte anvendes til at kontrollere flere af disse områder. Solihin og Eastman (2015) identificeret fire klasse af regler, som bundet i kompleksiteten af den granskning, der udføres.

Den simpleste klasse: **klasse 1** omfatter regler, som udelukkende kan udføres på baggrund af de data der direkte er tilgængelige i en BIM-model (eller anden datamodel).

Det er muligt at nogle af disse data er udledt af det BIM-værktøj, som modellen er skabt i, men i udgangspunktet skal de være direkte tilgængelige på objektet.

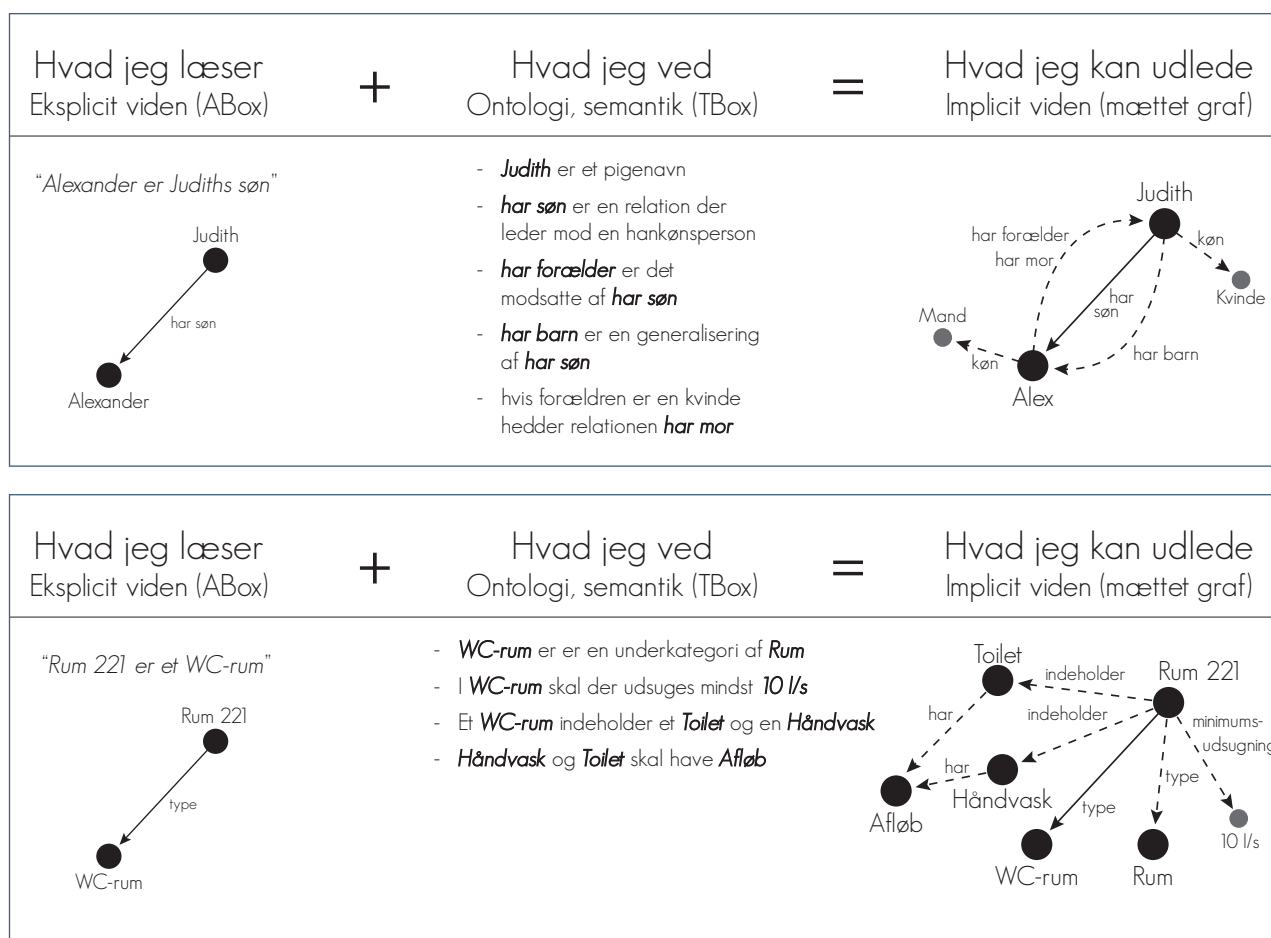
Klasse 2 bruges til at betegne regler, der bygger på data, der ikke nødvendigvis er eksplicit tilgængelige i modellen, men som kan udledes vha. logisk inferens / mætning (se figur 2). En afledt egenskab kan logisk afledes ud fra 1) de data der er eksplicit givet og 2) noget domæneviden. Eksempelvis kan det afledes at hvis et objekt tilhører klassen `IfcDoor`, så tilhører det også `CCS`-klassen `QQC`. Ligeledes kan man lave en klasse 2-regel, der selv beregner arealet af en dør ved at gange dørens højde og bredde. Reglerne bygger med andre ord på nogle af de implicite oplysninger i bygningsmodellen.

Klasse 3 handler om afledning af komplekse egenskaber som eksempelvis kræver simulering. Eksempelvis kunne man forestille sig en klasse 3 regel som benytter en trianguleringsalgoritme til at sikre at afstanden mellem røgdetektorer i flade lofter i gangarealer ikke overstiger 7 m.

Klasse 4 regler kan ikke blot identificere steder i modellen, hvor der potentielt er fejl og mangler, men kan også forklare hvorfor reglen ikke er overholdt og måske også give et forslag til en alternativ løsning.

- **Klasse 1 - Regler, der kræver en enkelt eller et lille antal eksplicite data**
"Findes der røgdetektorer i de rum hvor dette er et krav?"
- **Klasse 2 - Regler, der kræver enkle afledte egenskabsværdier**
"Hvad er afstanden mellem røgdetektor A og B? (afstand mellem centerpunkter beregnet)"
- **Klasse 3 - Regler, der kræver udvidet datastruktur**
"Er der korrekt afstand mellem røgdetektorer i lofter i gangarealer?"
- **Klasse 4 - Regler, der kræver bevisførelse**
Røgdetektor A og B er for langt fra hinanden og bør placeres således i stedet

Alene at de første to klasser er relativt enkle at implementere, men ikke desto mindre har de et enormt potentiale ift. at forbedre kvaliteten af bygningsmodeller (Solihin og Eastman 2015). Der findes ikke løsninger der implementerer klasse 3 og 4 i ARC-sammenhæng, selvom vi med generative design ser ad-hoc løsninger som gør brug af ARC-teknologierne. Programmet Solibri Model Checker er i stand til at bruge klasse 3-regler (eksempelvis til beregning af flugtveje), men det kræver en stor ekspertise idet de skal indarbejdes i programkoden, hvilket gør processen ufleksibel.



Figur 2: Denne figur forklarer hvad det vil sige at berige eller "mætte" en bygningsmodel med implicit viden. Den øverste del af figuren viser, hvordan vi mennesker hver dag udleder en masse implicit viden fra vores omgivelser, og den nederste del af figuren viser hvordan dette princip kan overføres til en bygningskontekst. Dette er muligt med den gren af kunstig intelligens (AI), som kaldes "description logics" (DL) og som blandt andet indgår i det semantiske web.

Semantiske webteknologier kan være vejen frem for ARC

Som det fremgår af ovenstående er der stadig en række udfordringer med at implementere ARC i byggeriet i stor skala. Der er behov for en ny tilgang til ARC, hvilket også fremhæves i et ph.d.-studie af Zhang (2019).

Zhang (2019) har gennem sit 5-årige Ph.d.-forløb arbejdet med ARC under den præmis at det skulle bygge på åbne standarder. Han forsøgte sig indledningsvist med en løsning bygget på open-source projektet BIMserver ved brug af gængse åbne BIM-standarder: IFC til at beskrive bygningsmodellerne, Model View Definitions og mvdXML til at repræsentere regelsæt og BIM Collaboration Format (BCF) til at rapportere problemer. Han oplevede dog denne løsning som uhensigtsmæssig og ikke-skalerbar af flere årsager, og gik i stedet over til at bruge semantiske webteknologier og linked data, og dermed opnåede han en langt større fleksibilitet. De semantiske webteknologier og linked data tillader et udvideligt udviklingsmiljø som er fordelagtigt ved udvikling af nye regelsæt og funktionaliteter, fordi det bygger på et generelt, alsidigt standard-sprog til at beskriver verden kaldet Ressource Description Framework (RDF). Zhang (2019) konstaterede at de løsninger, der er tilgængelige på markedet i dag, er utilstrækkelige - særligt i forhold til at kunne udføre geometrisk afhængige klasse 2-regler. Derfor udviklede han en udvidelse til det query sprog, som bruges til at lave informationsforespørgsler på det semantiske web, SPARQL. Hans udvidelse er særligt målrettet byggeriet og har fået navnet BimSPARQL. Zhang (2019) og mange andre forskere er enige om at semantiske webteknologier kan være vejen frem for ARC, og både buildingSMART og W3C-gruppen Linked Building Data arbejder aktivt på at udbrede kendskabet til teknologien. Shape Constraint Language (SHACL)² er en udvidelse til den tech stack der udgør det semantiske web, og den formaliserer ikke blot formulering af regler, men også rapportering af resultaterne. Der er ikke meget forskning der bruger den i en byggeri-sammenhæng, men de få studier der er, vidner om at der er potentiale (Stolk 2020, Oraskari 2021).

² <https://www.w3.org/TR/shacl/>

Systemopbygning

Overordnet kan et ARC-system ifølge blandt andre Solihin (2015) og Beach (2020) bygges op af fem elementer. I Tabel 1 beskriver vi de 5 elementer og kommer med eksempler på hvordan elementerne kan implementeres i praksis.

	Hvad indebærer et ARC-system?	Hvordan kan det implementeres i praksis?
1	Fortolkning af skrevne regler og konvertering til en form, der kan implementeres på computer: Hvordan skal reglerne forstås? ³	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 hvilke objekter, attributter og relationer, der er nødvendige, for at understøtte kontrollen. Hvilken delmængde af data er nødvendig for at kunne udføre kontrollen? ⁴	Modeller kan konverteres til RDF med eksisterende værktøjer ³ og forespørgsler på den relevante delmængde af data kan herefter ske med SPARQL ⁴ . Eventuelt skal der udføres yderligere opkvalificering af modeldata (geometriprocessering, specifikke relationer).
3	Udførelse af reglen, dvs. behandling af logikken i reglen, ved hjælp af betingelser, kombinatoriske betingelser og logiske spørgsmål. ⁵	Enten beskrives SPARQL queries til dette formål, eller hvis reglerne er beskrevet i SHACL processeres denne i et standardværktøj som PySHACL ⁵ .
4	Rapportering af resultaterne af regelkontrollen til de berørte parter på en sådan måde, at det bliver lettere at korrigere de identificerede problemer. ⁶	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 (Stolk 2020).

Tilbageværende løsningsrum

Overordnet bør man overveje følgende hovedpunkter når man udvikler ARC-løsninger:

- Nuværende kommercielle løsninger tillader ikke umiddelbart udvikling af nye regelsæt som er baseret på åbne standarder. Dette betyder at de nuværende løsninger på markedet er direkte afhængige af proprietær software (Greenwood et al. 2010, Zhang 2019).
- Kløften mellem de data, der skal kontrolleres og reglernes ordlyd kræver at ARC-applikationer er i stand til at finde implicite data ud fra de oplysninger, der eksplicit er til rådighed (klasse 2 og 3) (Solihin og Eastman 2015).

³ <https://www.w3.org/TR/shacl/>

⁴ 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/>

⁵ <https://www.w3.org/TR/sparql11-query/>

⁶ <https://github.com/RDFLib/pySHACL>

- Reglerne bør tilpasses efter projektstadiet (LoD). Det kræver en fleksibel og lettilgængelig tilgang til oprettelse af regler, og det egner semantiske webteknologer sig godt til (Solihin og Eastman 2015, Beach 2020, Zhang 2019).
- ARC kræver en klar forståelse af semantikken i de undersøgte data og i de regler der ønskes anvendt, og med semantiske webteknologier kan kompleksiteten bringes ned ift. hvad der er tilfældet med IFC (Solihin og Eastman 2015, Zhang 2019).
- Der er identificeret en mangel på fælles åbne standarder og et fælles sprog for regelbestemmelser og ARC-håndtering, men SHACL kan potentielt være et element i den rigtige retning (Greenwood et al. 2010, Beach 2020, Solihin og Eastman 2015, Stolk 2020, Oraskari 2021).

De næste faser af BART vil undersøge om SHACL og semantiske webteknologier er velegnede til ARC i byggeriet.

Ethvert byggeprojekt skal kvalitetssikres. Dette sker i dag gennem en manuel granskningsproces.

Automated Rule Checking (ARC)

automatiserer hele eller dele af granskningsprocessen

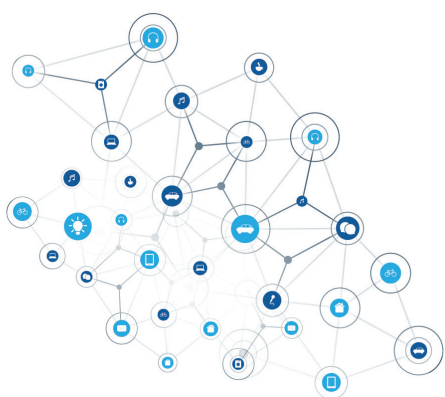


Lever byggeprojektet op til de prædefinerede krav?

Projekteringsfejl i dansk byggeri beløber op imod **40 mia. kr. årligt**
Som potentielt kan afhjælpes med ARC

Automatic Rule Checking (ARC) konverterer prædefinerede krav til regelsæt for byggeriet. Der findes 4 klasser af regler:

- **Klasse 1 - Regler, der kræver et lille antal eksplicite data**
"Findes der røgdetektorer i de rum hvor dette er et krav?"
 - **Klasse 2 - Regler, der kræver enkle afledte egenskabsværdier**
"Hvad er afstanden mellem røgdetektor A og B?"
 - **Klasse 3 - Regler, der kræver udvidet datastruktur**
"Er der korrekt afstand mellem røgdetektorer i lofter i gangarealer?"
 - **Klasse 4 - Regler, der kræver bevisførelse**
Røgdetektor A og B er for langt fra hinanden og bør placeres således i stedet
- } Lav kompleksitet men stort potentiale for økonomisk gevinst



Den nyeste forskning peger på at semantiske webteknologier, er vejen frem for ARC.

Med semantiske webteknologier kan ARC

- understøtte prægranskningsprocessen som i dag typisk er håndholdt.
- forbedre datakvaliteten ved at berige modellerne med implicit viden.
- sikre at regler er baseret på åbne standarder, der gør dem tilgængelige for alle byggeriets parter uanset softwareløsning.

Herved kan ARC i langt højere grad end i dag skaleres til give værdi for hele bygge- og anlægsbranchen.



