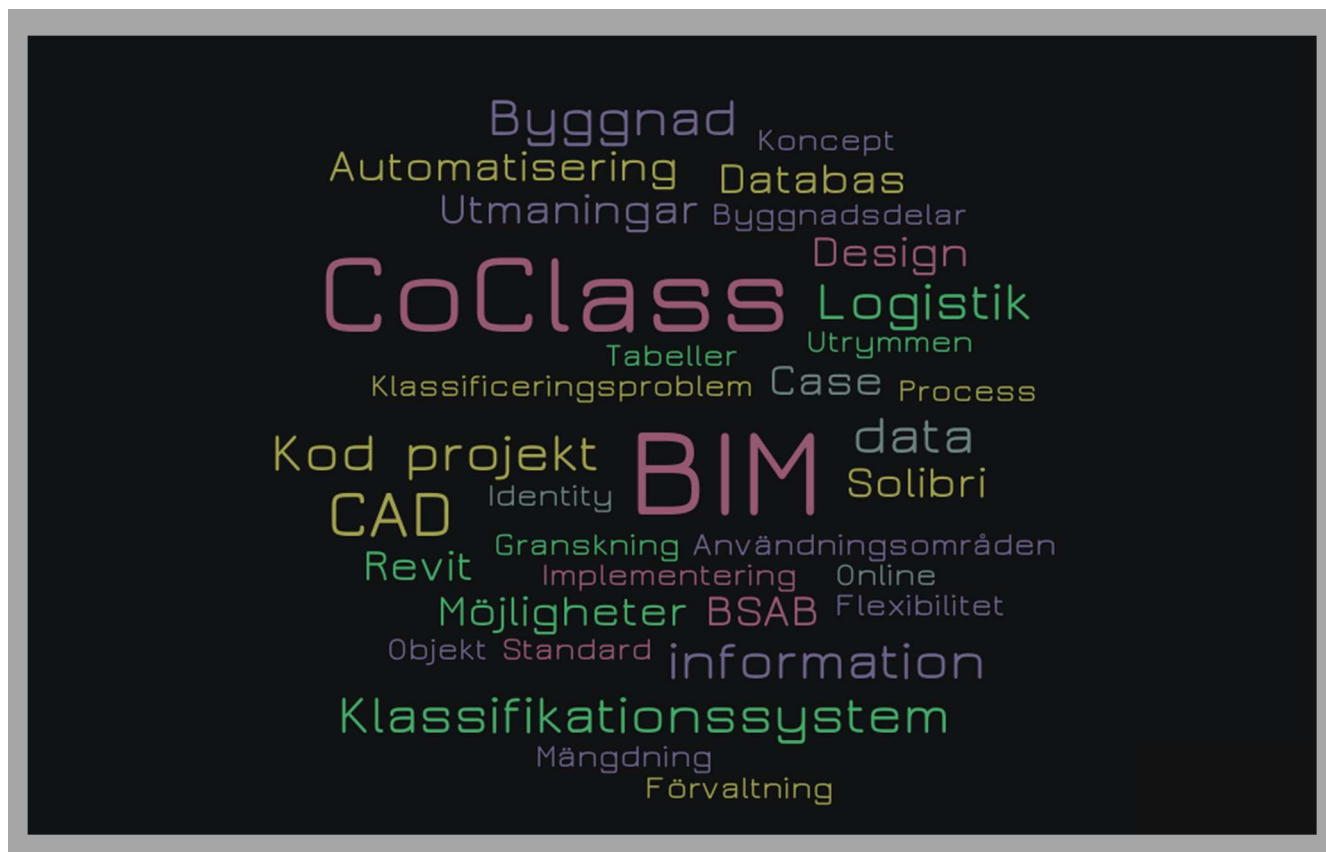




CHALMERS



Analys och tillämpning av CoClass

Examensarbete inom högskoleingenjörsprogrammet Byggingenjör

OLLE CARLSSON

Institutionen för arkitektur och samhällsbyggnadsteknik
Avdelningen för Construction Management

CHALMERS TEKNISKA HÖGSKOLA
Examensarbete BOMX03-17-42
Göteborg, Sverige 2017

EXAMENSARBETE BOMX03-17-42

Analys och tillämpning av CoClass

Examensarbete i högskoleingenjörsprogrammet

Byggingenjör

OLLE CARLSSON

Institutionen för arkitektur och samhällsbyggnadsteknik
Avdelningen för Construction Management
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, 2017

Analys och tillämpning av CoClass
Examensarbete i högskoleingenjörsprogrammet
Byggingenjör
OLLE CARLSSON

© OLLE CARLSSON, 2017

Examensarbete BOMX03-17-42 / Institutionen för arkitektur och
samhällsbyggnadsteknik, Chalmers tekniska högskola 2017

Institutionen för arkitektur och samhällsbyggnadsteknik
Avdelningen för Construction Management
Chalmers tekniska högskola
412 96 Göteborg
Telefon: 031-772 10 00

Omslag:
Figuren är skapad av författaren och inkluderar nyckelord gällande CoClass och BIM
– användning.

Institutionen för arkitektur och samhällsbyggnadsteknik
Göteborg 2017

Analys och tillämpning av CoClass

Examensarbete i högskoleingenjörsprogrammet

Byggingenjör

OLLE CARLSSON

Institutionen för arkitektur och samhällsbyggnadsteknik

Avdelningen för Construction Management

Chalmers tekniska högskola

SAMMANFATTNING

Det nya klassifikationssystemet CoClass är tänkt att under en snar framtid ersätta det idag använda systemet BSAB 96 på den svenska byggbranschen. Eftersom CoClass är nytt och ännu inte används i större utsträckning råder det brist på guider och texter gällande användning och implementering av detta nya system. Denna studie syftar därför till att undersöka och analysera begränsningar och möjligheter gällande CoClass. Studien inleddes med en litteraturstudie av tillgängliga dokument gällande CoClass samt BIM i stort. Själva undersökningen bestod sedan av en fallstudie där olika byggnadsdelar som klassificerades med hjälp av CoClass onlinebaserade databas användes. Vidare studier utfördes med hjälp av programvarorna Revit och Solibri Model Checker. Avslutningsvis finns en diskussionsdel där problem och möjligheter för CoClass diskuteras baserat på studiens resultat.

Case – studierna visade att CoClass är väl utrustat med varierade och välfyllda tabeller och bibliotek. På grund av detta verkar systemet vara fullt kapabelt, och alla valda byggnadsdelar kunde klassificeras på flera olika sätt. Däremot uppstod vissa oklarheter, främst gällande hur byggnadsdelarna bör kopplas till omgivande objekt och utrymmen. De efterföljande studierna i Revit och Solibri visade att Solibri kan sortera och granska enligt CoClass – kod som lades in i modellen via Revit. Det visade sig även att det teoretiskt sätt är möjligt att automatiskt generera de mindre unika delarna i en CoClass – kod.

För att bättre kunna implementera och använda CoClass verkar det vara fördelaktigt att dela upp koden i sina beståndsdelar när informationen matas in i en BIM – programvara som Revit. Denna metod innebär flera fördelar, däribland enklare ändring av information och bättre överblick. Dessutom blir arbetet i Solibri mycket mer effektivt och flexibelt om koden delas upp. En annan viktig slutsats som dragits är att möjligheten att automatiskt generera CoClass – kod skulle leda till att implementering av CoClass i ett projekt blir både billigare och mer effektivt. Det skulle även göra CoClass mer attraktivt för företag i branschen, och därmed öka branschens intresse av att använda CoClass i sin dagliga rutin.

Nyckelord:

CoClass, klassifikationssystem, BIM och automatisering.

Analysis and applications of CoClass

*Diploma Thesis in the Engineering Programme
Building and Civil Engineering*

OLLE CARLSSON

Department of Architecture and Civil Engineering
Division of Construction Management
Chalmers University of Technology

ABSTRACT

The newly developed classification system CoClass is intended to replace the currently used system BSAB 96 on the Swedish building industry within a reasonable amount of time. Since CoClass is newly developed and has yet to see much practical use, there is currently a shortage of texts and guides providing guidance and insight on how to use CoClass. Therefore, this paper aims to analyze the limitations and opportunities concerning the applications of CoClass. The first part of the study consisted of a literary study, focusing on all available documents regarding CoClass, as well as a couple of handbooks concerning BIM. After that a series of Cases – studies of various building parts were conducted by using CoClass's online – based database. Further studies were conducted by utilizing BIM – software programs Revit and Solibri Model Checker. Finally, the report ends with a discussion concerning the limitations and opportunities of CoClass. This discussion is based on the results of the study.

The Case – studies showed that CoClass is well equipped to handle most of the realistic situations that may occur when classifying the chosen building parts. However, some uncertainties did arise regarding how some of the building parts should be connected to surrounding spaces and objects. The following studies in Revit and Solibri showed that Solibri is capable of sorting objects and spaces by utilizing CoClass – code that was applied via Revit. It also revealed the possibility (in theory) to automatically generate the less unique parts of a CoClass – code.

When reviewing the results of the study, it seems like a solid way to improve the implementation and use of CoClass would be to split the code into individual parts when inserting the code into a BIM – software like Revit. Use of this method provides several advantages, easier updates and enhances clarity among them. This method also makes use of Solibri way more effective and flexible. Another important conclusion is that the possibility to automatically generate CoClass – code would lead to the implication of CoClass in a project becoming both cheaper and more effective. It would also make CoClass a more attractive option for companies in the industry, therefore increasing the industries desire to implement and use CoClass in their daily routine.

Key words:

CoClass, classification system, BIM, automation.

Innehåll

SAMMANFATTNING	I
ABSTRACT	II
INNEHÅLL	III
FIGURFÖRTECKNING	IV
FÖRORD	V
1 INLEDNING	1
1.1 Bakgrund och problembeskrivning	1
1.2 Syfte	1
1.3 Frågeställning	1
1.4 Avgränsningar	2
1.5 Metod	2
2 TEORETISK BAKGRUND	3
2.1 Konceptet BIM	3
2.1.1 Vad BIM är	3
2.1.2 Vad är inte BIM – teknologi?	4
2.1.3 Fördelar med implementering av BIM	4
2.1.4 Utmaningar och problem gällande BIM	7
2.2 Andra aktuella klassifikationssystem	7
2.2.1 BSAB 96	7
2.2.2 Cuneco Classification System	8
2.3 CoClass	8
3 RESULTAT: COCLASS - KLASSIFICERING AV BIM	9
3.1 Klassificering av projekt	9
3.1.1 Projektgemensamma delar i CoClass	9
3.1.2 Utrymmen i byggnaden	10
3.1.3 Identifiering av klassificeringsproblem – Utrymme & Projekt	10
3.2 Klassificering av byggnadsdel i Revit	11
3.2.1 Dörrar	12
3.2.2 Fönster	14
3.2.3 Väggar	15
3.3 Användning av CoClass och information i Solibri	16
3.3.1 Information i Solibri	16
3.3.2 Interaktion mellan Solibri och CoClass	17
3.3.3 Uppdelning av kod i Solibri	18
3.4 Automatisering av kod	18
4 DISKUSSION	21
CHALMERS, Arkitektur och samhällsbyggnadsteknik, Examensarbete BOMX03-17-42	III

4.1	Uppdelning av kod	21
4.2	Automatisering av CoClass - kod	21
5	SLUTSATSER	22
6	FRAMTIDA STUDIER	23
7	REFERENSER	24

Figurförteckning

Figur 1: Referensbeteckning för ett mötesrum i JSP2.	9
Figur 2: Mötesrum i projekt JSP2.	11
Figur 3: Referensbeteckningar för dörr i JSP2.	12
Figur 4: Referensbeteckningar för fönster i JSP2.	14
Figur 5: Referensbeteckningar för en vägg i JSP2.	15
Figur 6: Uppdelad CoClass - kod i Solibri.	18
Figur 7: Relationer i Solibri.	19

Förord

Detta arbete är ett avslutande examensarbete motsvarande 15 högskolepoäng på Byggingenjörsprogrammet vid Chalmers Tekniska Högskola.

Jag vill tacka de trevliga personer från Zynka BIM AB som ställde upp på två givande och intressanta möten. Jag vill även tacka Mikael Johansson och Mikael Viklund Tallgren vid avdelningen för Construction Management på Chalmers Tekniska Högskola för värdefull hjälp under arbetets gång. Slutligen vill jag tacka min handledare Mattias Roupé vid avdelningen för Construction Management på Chalmers Tekniska Högskola för hans utmärkta vägledning och stöd.

Göteborg september 2017
Olle Carlsson

1 Inledning

Det nya digitala klassifikationssystemet CoClass kommer inom de närmsta åren implementeras mer och mer i den svenska byggbranschen, men det är inte helt självklart hur denna implementering ska gå till och exakt hur CoClass ska användas. Författaren fick därför i uppdrag av avdelningen för Construction Management på Chalmers Tekniska Högskola att utföra en studie av CoClass, utvärdera hur systemet fungerar och om några problem och utmaningar finns.

1.1 Bakgrund och problembeskrivning

Den svenska byggbranschen använder sig idag av klassifikationssystemet BSAB 96. Branschprojektet BSAB 2.0 har under en tid jobbat på det nya klassifikationssystem CoClass, som så småningom kommer att ersätta BSAB 96 på den svenska byggbranschen. Jämfört med BSAB 96 är CoClass utrustat med större och mer varierade tabellbibliotek, vilket ger CoClass en större flexibilitet jämfört med BSAB 96. Man hoppas alltså på att CoClass, i sinom tid, ska medföra stora förbättringar i den svenska byggbranschen.

Eftersom CoClass fortfarande är i ett relativt tidigt stadie finns ont om relevant litteratur och guider som kan hjälpa branschen att sätta sig in i hur CoClass fungerar. Därför är en kompletterande undersökning som kan hjälpa till att tydliggöra och utveckla CoClass möjligheter och utmaningar av stor relevans.

1.2 Syfte

Detta examensarbete syftar till att undersöka och analysera begränsningar och möjligheter i nya klassifikationssystemet CoClass. Undersökningarna gäller CoClass som system samt dess implementering och användning i BIM-programvaror så som Revit och Solibri. Det syftar även till att skapa en relevant rapport som, tillsammans med existerande litteratur, kan ge en bättre överblick över CoClass användningsområden och möjligheter.

1.3 Frågeställning

Studiens huvudfokus ligger på att undersöka begränsningar och möjligheter i CoClass med hjälp av olika metoder. För att matcha detta syfte har följande frågeställningar formulerats:

Vilka begränsningar och möjligheter finns i CoClass?

Hur kan man gå tillväga för att implementera CoClass i tredimensionella ByggnadsInformationsModeller?

Är det möjligt att automatiskt generera en kodsträng som följer CoClass standard från ByggnadsInformationsModeller?

1.4 Avgränsningar

- Tester och undersökningar utförs enbart på objekt och utrymmen som återfinns i arkitektmodellen för projekt JSP2.
- Ett antal objekt och utrymmen väljs ut – omöjligt att studera hela arkitektmodellen.
- Studierna görs uteslutande med avseende på CoClass.

1.5 Metod

Studien baseras främst på följande delar; En inledande litteraturstudie, ett antal generella tester för att komma in i hur CoClass fungerar, samt ett antal fallstudier. Avslutningsvis gjordes tester i Solibri angående möjligheter för automatisering av CoClass – kod.

Den inledande litteraturstudien syftade till att lägga in grund för arbetet genom att granska konceptet BIM, och bestod av *A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors* (Eastman, 2011) och *BIM and Construction Management - Proven Tools, Methods, and Workflows* (Hardin, 2009). Litteraturstudien gick även in på Svensk Byggtjänsts publicerade dokument gällande CoClass. De använda källorna har granskats kritiskt och bedömts vara passande och trovärdiga.

Testerna av CoClass utfördes främst genom trial-and-error – metoden, och de genomfördes främst i programvarorna Revit och Solibri Model Checker. Inledande generella tester ledde till en fallstudie av de utvalda fokusobjekten i projektet. Resultaten av fallstudien användes sedan för att studera hur CoClass – kod inlagd i ett Revit-projekt kan granskas i Solibri. Slutligen gjordes en teoretisk uppskattning av möjligheten att automatiskt generera CoClass – kod baserat på resultaten från studierna i Revit och Solibri.

2 Teoretisk bakgrund

För att kunna förstå fördelar, nackdelar och problem hos CoClass behövs en övergripande förståelse för studiens teoretiska bakgrund. Detta kapitel kommer därför att redogöra för konceptet BIM i stort. CoClass föregångare kommer att beskrivas, samt slutligen en redogörelse över vad exakt CoClass är, hur det fungerar och vad det ska användas till.

2.1 Konceptet BIM

Detta kapitel ger en översikt över begreppet ByggnadsInformationsModellering (BIM). Bland annat behandlas vad som kännetecknar BIM, hur man kan implementera BIM i ett projekt inom byggbranschen samt de problem och utmaningar som kan uppstå då man jobbar med BIM.

2.1.1 Vad BIM är

Enligt Eastman (2011), definieras ByggnadsInformationsModellering (BIM) som en modelleringsteknologi med tillhörande aktiviteter som kan skapa en byggnadsmodell, och sedan kommunicera med samt analysera modellen. Modellen i fråga kan sedan utrustas med information gällande bland annat dimensioner, tillverkare och typ.

Eastman (2011) definierar ByggnadsInformationsModeller (BIM) med hjälp av följande punkter:

- Byggnadsdelar redovisas i digital form med hjälp av objekt som innehåller grafisk information samt de specifika egenskaper som motsvarande byggdel skulle kunna ha i verkligheten. Båda dessa parametrar kommer dessutom med regler som gör att byggnadsdelen bibehåller en intelligent geometri och byggbarhet vid redigering.
- Komponenter i en modell innehåller data som beskriver hur de förhåller sig till varandra. Tack vare denna data kan man enkelt ta ut en mängdning eller göra en energianalys direkt i modellen.
- Ändringar som görs i ett objekt hänger automatiskt med till alla vyer och lägen som objektet är en del av. Man kan alltså vara säker på att ändringen implementeras överallt i modellen.
- Samtliga vyer i modellen är koordinerade och kan plockas ur systemet separat för att sedan slås ihop till en enhetlig byggnadsmodell.

Definitionen ovan beskriver vad som oftast kallas för BIM i 3D-format. I boken "BIM and Construction Management - Proven Tools, Methods, and Workflows" tar Hardin (2009) upp hur man kan gå tillväga för att lägga på en fjärde och en femte dimension i sin BIM – modell. 4D innebär i detta fall att man integrerar en produktionstidplan i modellen, vilket ger möjlighet att få fram en tidplan som hänger ihop med modellens objekt. 5D innebär att man förutom produktionsplan även integrerar kostnadsdata för objekt i modellen. Med hjälp av dessa data kan man ur modellen få fram en preliminär kostnadskalkyl, vilket kan vara till stor hjälp vid ett tidigt skede i projektet.

2.1.2 Vad är inte BIM – teknologi?

Eastman (2011) tar även upp ett antal exempel på modelleringsalternativt som inte kan räknas som BIM – teknologi. Detta är särskilt relevant att nämna eftersom det är enkelt att dra slutsatsen att alla mjukvaror och tekniker som använder sig av visuell modellering även använder sig av BIM, när så inte alltid är fallet. Eastman (2011) använder sig av följande exempel:

- 3D – modeller som använder sig av visualisering men inga (eller få) objekttegenskaper kopplade till de visuella objekten. En sådan modell kan inte användas för att göra analyser eller studier av något slag, utan dess värde ligger enbart i det man kan se med blotta ögat. Ett exempel på programvara som faller inom denna kategori är SketchUp, som är ett bra program för ren modellering men inte så mycket annat.
- Modeller utan stöd för anpassning. Modeller av dessa slag kan använda sig av objekt, men saknar förmågan att enkelt ändra objektets placering och / eller parametrar. På grund av detta krävs det mycket eget arbete för att göra något så enkelt som att flytta objektet från en plats till en annan. Dessutom saknar denna typen av modell de "intelligenta regler" som nämndes i kapitel 2.1.1, vilket innebär att modellen inte vet hur ett objekt bör se ut för att vara funktionellt. Med andra ord löper man en mycket högre risk att rita något som sedan inte är konstruktionsmässigt korrekt.
- Modeller som använder sig av flera separata CAD – referensfiler i 2D för att definiera en byggnad. På grund av att det är omöjligt att garantera att den slutgiltiga modellen är genomförbar får dessa modeller inte räknas som ByggnadsInformationsModeller.
- Modeller när det är möjligt att göra ändringar i ett objekt som inte direkt återspeglas i alla vyer som objektet finns i. Det är omöjligt att garantera att dessa modeller är enhetliga, eftersom fristående ändringar är möjliga, och de får således ej klassas som ByggnadsInformationsModeller.

2.1.3 Fördelar med implementering av BIM

BIM- teknologin är fortfarande i ett förhållandevis tidigt stadie ute i branschen. Trots detta ger användningen av BIM redan nu många fördelar som sparar både tid och pengar. Dessa fördelar kommer med största sannolikhet bara att bli fler och fler i takt med att utvecklingen av BIM fortsätter.

2.1.3.1 Fördelar under förstudie

Innan man som byggherre anlitar en arkitekt eller liknande är det enligt Eastman (2011) viktigt att på något sätt ta reda på om byggnaden i fråga ligger inom byggherrens avsedda budget och tidsschema. Ett snabbt och billigt sätt att få reda på detta är att skapa en BIM-modell som man sedan kopplar till en kostnadsdatabas. Om en sådan kostnadsundersökning ger ett positivt resultat kan man gå vidare till en arkitekt med relativt låg risk för att behöva betala för ett arkitektjobb som man sedan inte kan genomföra.

Vill man i ett tidigt skede få en inblick i hur en slutlig modell kommer att se ut kan en schematisk modell vara till stor hjälp. I den schematiska modellen kan man bland annat avgöra om byggnaden kommer att uppnå de krav som ställs på

funktionalitet och hållbarhet. Om dessa krav inte uppnås kan man vid ett tidigt skede vidta åtgärder för att säkerställa att den slutliga produkten kan klara kraven (Eastman, 2011).

2.1.3.2 Fördelar under projektering

En 3D - modell som skapas i ett BIM - program är alltid enhetlig. Det innebär att man kan vara säker på att det man ser i en vy även återspeglas i andra vyer, och att alla ändringar som görs kommer att implementeras i hela modellen. Enligt Eastman (2011) är detta en stor fördel eftersom man inte behöver kontrollera och redigera varje enskilt vy så fort en ändring ska implementeras. En annan fördel är att man tillgång till en användbar 3D – modell i ett tidigare skede i modellen, utan att behöva lägga tid och pengar på att slå ihop ett antal 2D – ritningar.

Vidare kan man med hjälp av BIM få till ett tidigare och mer effektivt samarbete mellan de olika disciplinerna i ett projekt. Genom att med jämna mellanrum slå samman de olika disciplinernas modeller kan man i god tid upptäcka bland annat designfel och objektskrockar i projektplanen. På grund av att dessa problem är oerhört svåra att upptäcka genom att jämföra 2D – ritningar kostade liknande fel tidigare stora summor pengar, då de ofta inte upptäcktes förrän under produktionen, men med hjälp av BIM kan man nu undvika dessa extra kostnader. (Eastman, 2011).

En annan stor fördel är att man när som helst under projekteringen kan ta ut mängder och ytor ur modellen. Med hjälp av denna information kan man, med hjälp av en kostnadsdatabas, göra en preliminär uppskattning av projektets kostnad. Hur exakt denna uppskattning blir beror till stor del på vilket skede i projekteringsfasen man befinner sig i. I ett tidigt skede har man i stort sätt enbart tillgång till grövre mängder, exempelvis areor, antal parkeringsplatser/lägenheter, eller kostnad per enhet. En uppskattning gjord i detta stadie blir naturligtvis ganska grov och osäker, men man får ändå en uppfattning av vad den slutliga kostnaden kan bli. När modellen sedan blir mer detaljerad kan man gå in och koppla varje enskilt objekt till en specifik kostnad, vilket givetvis ger en mycket bättre uppskattning. (Eastman, 2011).

Slutligen nämner Eastman (2011) att man med hjälp av olika analysverktyg kan använda sin BIM – modell för att göra preliminära energianalyser. Jobbar man med 2D – ritningar tar man inte hjälp av energianalyser förrän sent i designprocessen, på grund av den höga kostnaden, och då enbart för att kontrollera att man kan möta de krav som ställs på byggnaden. Med hjälp av BIM kan man göra löpande energianalyser i takt med att arbetet fortskrider, och baserat på resultaten göra informerade ändringar för att få en så energieffektiv byggnad som möjligt.

2.1.3.3 Fördelar under produktion

En effektiv användning av BIM – teknologi kan ge stora fördelar under produktionen av en byggnad. Bland annat kan man överföra sin 3D – modell till en BIM – programvara som kan datorstyra och prefabricera komponenter

baserat på modellen. Ett krav för detta är att 3D – modellen håller en tillräckligt hög detaljnivå. Denna typ av prefabricering är enligt Eastman (2011) idag standard när det gäller prefabricering av stålkomponenter och plåtar. Man har även fått goda resultat inom bland annat gjutna komponenter och glaselement. Med hjälp av denna typ av prefabrikation kan man göra processen enklare samtidigt som man sparar både tid och pengar.

Genom att koppla en 3D – modell till en konstruktionsplan går det att få fram en så kallad 4D – modell. Denna modell kan användas för att simulera konstruktionsprocessen. Denna teknologi gör det alltså möjligt att mata in valfritt datum i konstruktionsplanen och få ut hur byggplatsen ser ut vid sagda datum. (Förutsatt att man är i fas med konstruktionsplanen). Det är även möjligt att göra en simulering som visar byggförloppet dag för dag. Med hjälp av dessa simuleringar går det att få en överblick över hur byggprocessen fortskrider, och ger även möjligheten att upptäcka svårigheter och/eller förbättringar som är svåra att upptäcka från pappersritningar, gällande exempelvis utrustning, byggmaterial och förvaringsmöjligheter. (Eastman, 2011).

En effektiv användning av BIM – teknologi kan vara till stor hjälp när man använder sig av lean construction. Lean construction går ut på att effektivisera byggprocessen genom att man kontinuerligt får leveranser av material som monteras så snart som möjligt. Denna levereringsteknik kallas ibland just-in-time. Man måste även se till att man har tillräckligt med personal på plats för att kunna ta emot och montera det levererade materialet. Användning av lean construction kan spara både tid och pengar genom en ökad effektivitet, förutsatt att man kan få det att fungera ordentligt. Sparad plats på grund av lägre krav på materialförvaring är en annan fördel. Det kan dock vara svårt att få en sådan planering att gå ihop, och det är här BIM kommer in. Ur en BIM – modell kan man enkelt få ut bland annat materialåtgång för de olika processerna, vilket gör att en lean construction – planering kan genomföras med större precision. (Eastman, 2011).

Slutligen kan man även under konstruktionen dra nytta av BIM – teknologins smidiga hantering av designändringar, kollisionsskontroller och liknande. Detta går att läsa mer om i kapitel 2.1.3.2.

2.1.3.4 Fördelar i förvaltningsstadiet

Den största fördelen med BIM i förvaltningsstadiet är att man som förvaltare har tillgång till en dynamisk, uppdaterad modell som visar alla konstruktioner, system och andra objekt som finns i byggnaden. En sådan modell kan vara till stor hjälp vid exempelvis till- eller ombyggnad, eftersom man har en exakt representation av byggnaden att utgå ifrån. Om något system i byggnaden skulle gå sönder kan man dessutom gå in i modellen och få en värdefull överblick, vilket underlättar vid reparationer av större, mer komplicerade system. (Eastman, 2011).

Hardin (2009) nämner även att 60-85% av en byggnads totala livscykelkostnad är drift- och underhållskostnader under förvaltningen. Förhoppningen är att

man i framtiden ska kunna använda BIM för att minska denna kostnad och skapa en mer effektiv byggnadsförvaltning.

2.1.4 Utmaningar och problem gällande BIM

Användning av BIM underlättar idag inom många områden i byggbranschen, och både antal användningsområden och omfattning kommer säkerligen att öka i framtiden. För att denna utveckling ska kunna ske krävs det dock att stora förändringar görs, både i branschen som helhet men även i hur enskilda företag arbetar.

Eastman (2011) tar upp ett antal utmaningar för BIM. Den första utmaningen som Eastman (2011) tar upp är problem som kan uppstå vid samarbete mellan olika företag. Exempelvis kan problem och konflikter uppstå när parter som använder olika programvaror ska samarbeta inom samma projekt. Om filerna inte är konverterbara mellan de olika parternas programvaror kan extra kostnader uppstå. Ett annat exempel kan vara att ett projekt innehåller en arkitekt som jobbar med pappersritningar och en konstruktör som jobbar i BIM. Då måste konstruktören först skapa en grundmodell som en annan arkitektfirma kanske redan har skapat. Problem som dessa kan skapa konflikter och potentiellt leda till ett sämre utfört arbete.

Ett annat problem som nämns av Eastman (2011) är att det kan uppstå oklarheter om vilka parter som äger de olika modeller som används i ett projekt. Det kan även vara oklart vem som ska bekosta modellerna, och vem som är ansvarig för att modellerna motsvarar verkligheten, eftersom parterna delar på användningen av modellerna.

Eastman (2011) går även igenom de förändringar som enskilda företag kommer att behöva göra för att kunna möta framtidens BIM – behov. Förutom att skaffa modern programvara och personal som kan hantera den, behöver företagen göra förändringar inom infrastruktur och organisation. Det kommer att både tid, kraft och energi för det genomsnittliga företaget att hänga med i utvecklingen, men trots detta menar Eastman (2011) att det i längden kommer att vara till företagets fördel.

2.2 Andra aktuella klassifikationssystem

Det finns många andra klassifikationssystem, men detta arbete kommer att ta upp två andra system: BSAB och Cuneco Classification System.

2.2.1 BSAB 96

Det klassifikationssystem som för närvarande används i den svenska byggbranschen heter BSAB 96, och det har använts sedan 1970 – talet. Huvudsyftet med BSAB är enligt Svensk Byggtjänst (2017a) att identifiera, dela upp och sortera information. Trots att BSAB 96 har varit en del av den svenska byggbranschen i snart 50 år har det fått kontinuerliga uppdateringar vilket har gjort att det fortfarande är aktuellt i dagens läge. Svensk byggtjänst har genom

branschprojekt BSAB 2.0 utvecklat det nya digitala klassifikationssystemet CoClass. CoClass är även känt som den nya generationen BSAB, och kommer så småningom att ersätta BSAB 96 i den svenska byggbranschen. (Svensk Byggtjänst, 2017a).

2.2.2 Cuneco Classification System

Cuneco Classification System är det nyutvecklade digitala klassifikationssystem som används i Danmark. Det används hela vägen under hela projektet från start till slut, och används även i projektets underhållningsfas. Cuneco är alltså Danmarks motsvarighet till CoClass. Skovgaard (2013) nämner att Cuneco främst fokuserar på att utveckla sin standard inom fyra områden; klassifikation, property data, informationsnivåer och måttregler. Cuneco finns tillgängligt för den danska byggbranschen via en databas som tillhandahåller alla de verktyg och tabeller som Cuneco tillhandahåller. (Skovgaard, 2013).

2.3 CoClass

CoClass är, som nämndes i kapitel 2.2.1, resultaten av BSAB 2.0, ett branschgemensamt projekt som inkluderade en betydande del av aktörerna på den svenska byggbranschen. Svensk Byggtjänst (2017b) beskriver CoClass på följande vis; *"CoClass är det nya digitala klassifikationssystemet för all byggd miljö i Sverige med potential att för all framtid förändra bygg- och förvaltningssektorn. CoClass bedöms kunna bidra till miljardbesparingar genom att utgöra plattform för effektivare kommunikation och "asset management".*" För att uppnå denna beskrivning beskriver Svensk Byggtjänst (2017b) att CoClass syftar till att uppnå följande sex värden.

- Digitalt – ger dig platshållare för digital information
- Gemensamt – samma språk och samma struktur kan användas oavsett i vilken situation CoClass används.
- All byggd miljö – CoClass kan täcka alla typer av byggd miljö.
- Hela livscykeln – CoClass är utrustat för att täcka varje projekts behov genom hela dess livscykel, från planering till rivning.
- Internationellt - CoClass finns i två språk, Svenska och Engelska, och är utformat med hänsyn till internationella standarder.
- Framtidssäkert – CoClass använder sig av öppna och flexibla strukturer som är baserade på funktion, vilket innebär att systemet kan anpassas efter framtida utvecklingar i branschen.

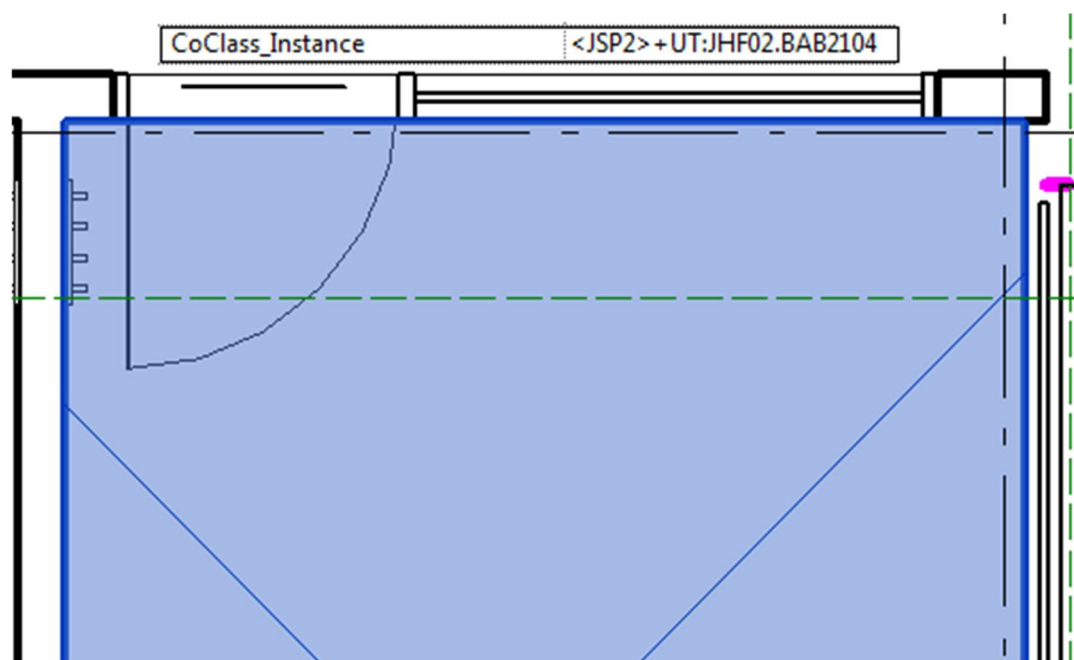
3 Resultat: CoClass - klassificering av BIM

För att kunna genomföra relevanta tester på CoClass referensbeteckningar behöver man applicera koden på realistiska målobjekt. Alla objekt som testas i denna rapport är tagna från aktuella BIM – modeller från det pågående projektet Johanneberg Science Park 2, även kallat JSP2. All kod är tagen från CoClass onlinebaserade databas (Svensk Byggtjänst, 2017c).

3.1 Klassificering av projekt

Tabellerna i CoClass har en öppen och flexibel struktur, vilket innebär att en referensbeteckning tillhörande ett visst objekt kan ha olika utseenden och strukturer (Svensk Byggtjänst, 2017c). Svensk Byggtjänsts (2016b) dokument Appendix A1 – Referensbeteckningar och exempel tar upp tre strukturer; Funktionsorienterad, produktionsorienterad och placeringsorienterad struktur. Vid användning av funktionsorienterad struktur utgår man från hela systemets funktion och går sedan ner i mindre komponenter och system, även kallat top-down-metoden. Vid en produktionsorienterad struktur utgår man istället från komponenten och jobbar sig upp genom systemen, även kallat bottom-up-metoden. Slutligen är den placeringsorienterade strukturen mer flexibel, och därmed inte begränsad till någon av ovanstående metoder. När de gäller de delar av referensbeteckningen som hänvisar till de större systemen har de olika strukturerna, trots sina olikheter, mycket gemensamt.

3.1.1 Projektgemensamma delar i CoClass



Figur 1: Referensbeteckning för ett mötesrum i JSP2.

Det finns ett antal referenser i ett projekt som kommer att återfinnas i många eller alla referensbeteckningar som skrivs. En av dessa är *toppnoden* som anger hela systemet, och betecknas med hjälp av vinkelparenteser. I figur 1 kan man se <JSP2> inom vinkelparenteser, vilket alltså visar på att byggnaden JSP2 är det totala systemet.

Andra referenser som kan anses vara mer eller mindre projektgemensamma är de grövre indelningarna av byggnaden, som exempelvis zoner eller våningsplan. Till skillnad från toppnoden, som alltid anges, är zon och/eller våningsplan något man tar med när det är relevant för referensbeteckningen i fråga. I figur 1 ovan används *+UT:JHF02* för att beskriva att objektet i fråga befinner sig i ett utrymme som innefattar hela det andra våningsplanet, där UT betyder att det är ett utrymme och JHF02 betyder att våning 2 är utrymmet som hänvisas. En mer utförlig förklaring över hur själva koden är uppbyggd kommer i följande stycken.

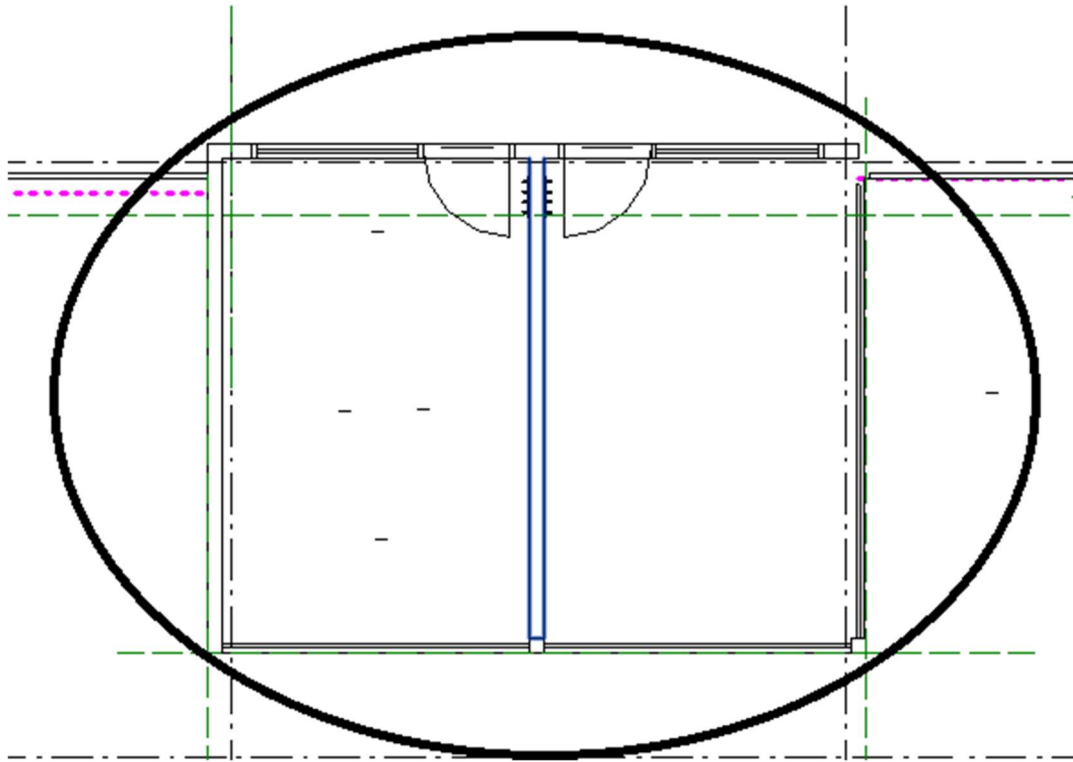
3.1.2 Utrymmen i byggnaden

När det gäller specifika utrymmen i en byggnad, exempelvis ett rum, får man helt enkelt leta upp den typ av rum man behöver i CoClass tabeller. I figur 1 ovan betecknar *BAB2104* ett mötesrum med löpnummer 2104. Ett färdigt system för löpnummer är inte något som CoClass tillhandahåller, då det är något som måste anpassas för varje individuellt projekt. För att få den totala referensbeteckningen i figur 1 kombineras mötesrummets referensbeteckning med passande placeringsorienterade referenser från kapitel 3.1.1.

3.1.3 Identifiering av klassificeringsproblem – Utrymme & Projekt

Inga större problem verkar uppstå när kod skrivs för utrymmen och projektgemensamma delar enligt CoClass. Koden i figur 1 till exempel beskriver ett utrymme (*BAB2104*) som befinner sig i ett annat utrymme (*+UT:JHF02*), som innefattar hela det andra våningsplanet, som i sin tur befinner sig i projektet i fråga (*<JSP2>*). Rent kodmässigt verkar det inte finnas några alternativa sätt att skriva koden i figur 1, och det uppstår inte heller några oklarheter angående kodens betydelse.

3.2 Klassificering av byggnadsdel i Revit



Figur 2: Mötesrum i projekt JSP2.

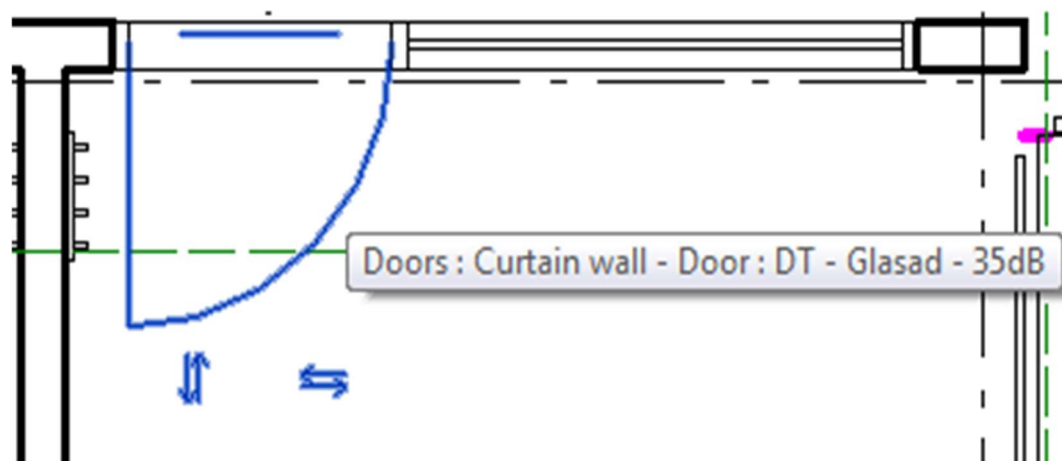
Till skillnad från de mer generella objekt och utrymmen som diskuteras i kapitel 3.1 kräver enskilda byggnadsdelar oftast längre, mer komplicerade referensbeteckningar. Det kommer även att krävas en mix av kod från de olika tabellerna, till skillnad från kapitel 3.1 som nästan enbart använde sig av koder för utrymmen. Det är först när man börjar använda sig av dessa längre referensbeteckningar som man inser hur flexibelt CoClass är; Det går att komma fram till samma resultat på ett antal olika sätt. Detta kapitel kommer att belysa olika metoder för att skriva referensbeteckningar för några vanligt förekommande objekt i en byggnad, och vilka fördelar och problem som uppstår. Objekt som undersöks här är alla plockade från rummen i figur 2, för att förmedla ett sammanhang för de olika byggnadsdelarna. En förklaring i textform kommer att ges för varje individuell kod.

3.2.1 Dörrar

CoClass – koden för en dörr kan, som visas nedan, se ut på ett antal olika sätt.

3.2.1.1 Klassificering – Dörrar

CoClass_Instance	<JSP2>+UT:JHF02.AD03-QQC2104%NSC22
CoClass_Instance	<JSP2>+UT:JHF02.EAD20-QQC2104%NSC22
CoClass_Instance	<JSP2>+UT:JHF02.BAB2104-QQC2104%NSC22



Figur 3: Referensbeteckningar för dörr i JSP2.

Vid undersökningen av kod för dörrar visade det sig finnas tre sätt att indikera en dörr. De tre resultaten ses i figur 3 tillsammans med dörren i fråga. CoClass tabeller är som sagt skapade för att ge stor flexibilitet, så det är fullt möjligt att det finns ytterligare sätt att indikera dörren i fråga än de tre sätt som listas här.

1. I den översta koden kopplas dörren till det övergripande systemet *via väggen den är monterad i*. I textform läses koden enligt följande: JSP2 har ett objekt med placering i innervägg AD03 på plan 2, nämligen en dörr med löpnummer 2104 av projektspecifik typ NSC22.
2. I den mellersta koden kopplas dörren till det övergripande systemet *via sin koppling till korridoren som ligger utanför mötesrummet*. Koden läses som följer: JSP2 har ett objekt med placering i korridorsrum 20 på plan 2, nämligen en dörr med löpnummer 2104 av projektspecifik typ NSC22.
3. I den nedersta koden kopplas dörren till det övergripande systemet *via sin koppling till mötesrummet den är kopplad till*. Koden läses som följer: JSP2 har ett objekt med placering i mötesrum 2104 på plan 2, nämligen en dörr med löpnummer 2104 av projektspecifik typ NSC22.

3.2.1.2 Identifiering klassificeringsproblem - Dörrar

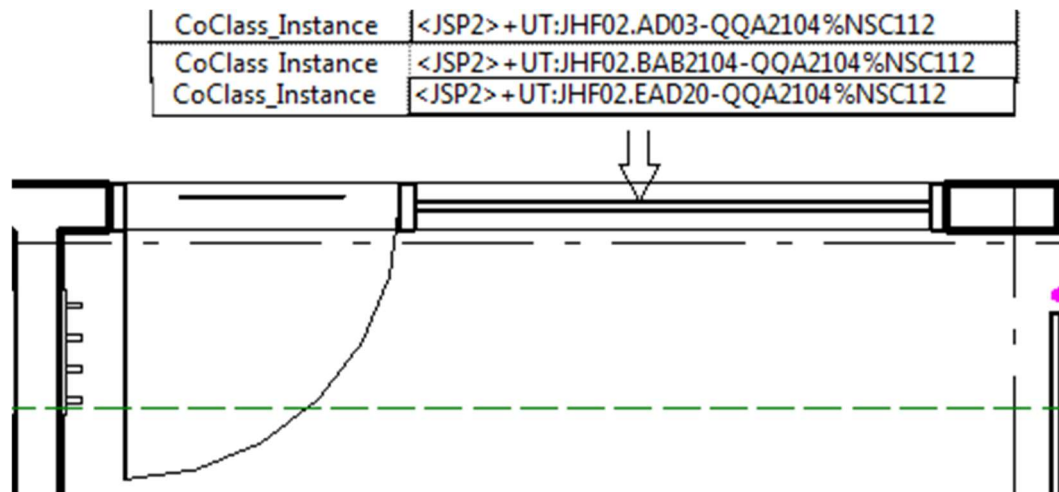
I detta kapitel granskas och diskuteras de olika kodningsalternativ som konstaterades i kapitel 3.2.1.1.

1. Kodningen i det första exemplet ovan är ett relativt bra sätt att specificera dörren i fråga, men det uppstår även vissa oklarheter. Väggen som dörren är kopplad till är relativt kort och har bara två dörrar installerade, men har man en längre vägg med fler kopplade dörrar kan det bli klumpigt att identifiera dörren i fråga. Det är självklart möjligt att gå in i modellen och identifiera dörren med hjälp av den tillhörande koden, men det är trots detta ett extra moment som krävs.
2. Punkt 2 och 3 från kapitel 3.2.1.1 har kopplat dörren till systemet på samma sätt, där enda skillnaden är vilket rum som dörren är kopplad till, så båda dessa punkter kommer att diskuteras här. Att koppla dörren till ett rum istället för en vägg gör att man enklare kan identifiera dörren (Rumsnamn & nummer är ofta utsatt i ritningen) men istället uppstår ett annat problem; Hur ska man bestämma vilket rum en dörr tillhör? En dörröppning är en portal mellan två rum, så det är absolut ingen självklarhet, även om det ibland kan se ut så. I exemplet från JSP2 ligger dörren i fråga mellan en korridor och ett mötesrum, så i det fallet är det rimligt att säga att dörren tillhör mötesrummet, eftersom rummet utanför är en "transportzon". Det verkliga dilemma uppstår om vi betraktar ett annat exempel; En dörr som leder från ett kontor till ett annat. En rationell metod vore att anta att en dörr hör till det rum som dörrbladet i fråga slår in i. Detta gäller dock inte i alla fall, så vissa dörrar, bland annat till toaletterum och liknande, har dörrblad som slår ut från rummet som de tillhör. Undersökningen har inte lyckats upptäcka någon metod för att lösa detta problem, men om det är något som går att arbeta runt, eller om det helt enkelt är en miss i CoClass system, återstår att se.

3.2.2 Fönster

Koden för ett fönster är relativt lik den för en dörr, men det är ändå vissa skillnader som uppstår.

3.2.2.1 Klassificering – Fönster



Figur 4: Referensbeteckningar för fönster i JSP2.

När man pratar om fönster så kan det tyckas självklart att de är placerade i en yttervägg, men det behöver inte vara fallet. I en kontorsbyggnad som JSP2 kan man ofta hitta fönster i anslutning till konferensrum och liknande, och detta är fallet med det mötesrum som är studiens fokus. Fönster i fasad är relativt okomplicerade, det går att koppla fönstret till fasadväggen, alternativt till rummet innanför, och behövs en rumstillhörighet finns det endast ett rum att välja på. Fönster i innerväggar stöter dock på ungefär samma problem som nämns för dörrar i kapitel 3.2.1. Gällande fönster har följande tre referensbeteckningar identifierats, se figur 4.

1. Den översta koden kopplar fönstret till väggen som det sitter i, på samma sätt som tidigare gjordes för dörren. Koden läses: JSP2 har ett objekt med placering i innervägg AD03 på plan 2, nämligen ett fönster med löpnummer 2104 av projektspecifik typ NSC122.
2. Den mellersta koden kopplar fönstret till systemet genom mötesrummet som fönstret ger inblick i. Koden läses: JSP2 har ett objekt med placering i mötesrum 2104 på plan 2, nämligen ett fönster med löpnummer 2104 av projektspecifik typ NSC122.
3. Den nedersta koden kopplar fönstret till korridoren utanför mötesrummet. Koden läses: JSP2 har ett objekt med placering i korridorsrum 20 på plan 2, nämligen ett fönster med löpnummer 2104 av projektspecifik typ NSC122.

3.2.2.2 Identifiering av klassificeringsproblem – Fönster

I detta kapitel granskas och diskuteras de olika kodningsalternativ som konstaterades i kapitel 3.2.2.1.

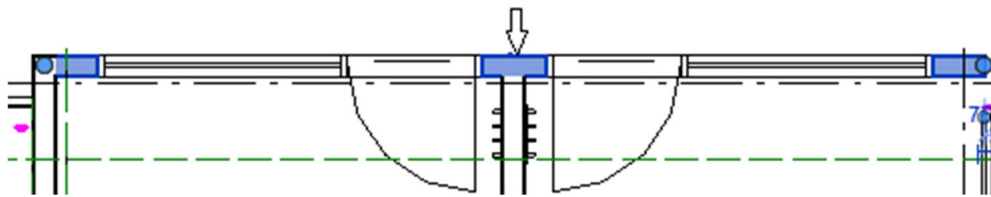
1. Koppling till vägg fungerar på samma sätt som i exemplet för dörrar som diskuteras i kapitel 3.2.1.2, dvs. att det fungerar, men att det ofta krävs att man går in i modellen och manuellt letar sig fram till objektet i fråga för att kunna identifiera enligt kodning.
2. Även här diskuteras punkt 2 och 3 från kapitel 3.2.2.1 tillsammans, då enda skillnaden är vilket rum fönstret är kopplat till. Samma diskussion och exempel som tas upp i kapitel 3.2.1.2, punkt 2 gäller även här, förutom att det inte finns något dörrblad att gå efter.

3.2.3 Väggar

Koden för väggar skiljer sig från dörrar och fönster i och med att man inte behöver koppla väggen till ett "värdeobjekt".

3.2.3.1 Klassificering – Väggar

CoClass_Instance	<JSP2>+UT:JHF02-AD03.21%GSA52
CoClass_Instance	<JSP2>+UT:JHF02.BAB2104-AD03.21%GSA52
CoClass_Instance	<JSP2>-AD01.4%GSA51



Figur 5: Referensbeteckningar för en vägg i JSP2.

Till skillnad från fönster och dörrar, som i de allra flesta fall har liknande form och funktion, har väggar mycket större variation. Det kan vara allt från en yttervägg som spänner över en hel fasad, till en kort innervägg som avgränsar ett förrådsutrymme. Detta, tillsammans med det faktum att en vägg är ett "fristående" objekt, gör att dess referensbeteckning utformas på ett annat sätt än vid de tidigare exemplen. Under försöken har följande tre referensbeteckningar identifierats:

1. Det övre exemplet listar helt enkelt *vilken våning väggen ligger på*, en metod som alltid går att applicera på en vägg, oavsett form och funktion. Koden läses: JSP2 har ett objekt med placering på plan 2, nämligen en innerväggskonstruktion med löpnummer 21 av projektspecifik typ GSA52.
2. Det mellersta exemplet kopplar *väggen till rummet det innesluter*. Koden läses: JSP2 har ett objekt med placering i mötesrum 2104 på plan 2, nämligen en innerväggskonstruktion med löpnummer 21 av projektspecifik typ GSA52.

3. Det nedersta exemplet gäller inte väggen i exemplet, utan är snarare ett exempel på hur en yttervägg eller en bärande vägg skulle kunna betecknas. Koden läses: JSP2 har en ytterväggskonstruktion med lönnummer 4 av projektspecifik typ GSA51.

3.2.3.2 Identifiering av klassificeringsproblem - Väggar

I detta kapitel granskas och diskuteras de olika kodningsalternativ som konstaterades i kapitel 3.2.3.1.

1. Alternativ 1 är, som nämndes i kapitel 3.2.3.1, ett generellt alternativ som alltid går att applicera. Rent positionsmässigt får man dock enbart ut vilket plan väggen tillhör, men information om position i planet saknas. Det innebär att en manuell kontroll av modellen krävs för att identifiera vilken vägg som indikeras.
2. Alternativ 2 kopplar alltså väggen till rummet som innesluts. En sådan koppling är dock bara realistisk i vissa situationer, framför allt i de fall där man har en innervägg som enbart innesluter ett enda rum. Väggen som är kopplad till mötesrummet i figur 5 kopplar även till mötesrummet bredvid, så för väggen i exemplet är detta alternativ inte att föredra, då det kan vara missledande. Det kan som sagt fungera om väggen i fråga enbart är ansluten till ett enda rum.
3. Alternativ 3 gäller som tidigare nämnt ytterväggar och bärande väggar, och det finns därför varken anledning eller möjlighet att applicera det på väggen i figur 5. Det är trots det ett exempel som är värt att ta upp. Skillnaden mellan en yttervägg och en innervägg är att ytterväggen skulle kunna anses sträcka sig över flera plan, och därmed kan inte en plantillhörighet anges. Samma situation gäller för vissa bärande väggar i projektet, då även dessa sträcker sig över flera plan. Resultatet blir att den enda information som kommer med i referensbeteckningen är toppnod och specifikationer för väggen i fråga. Ytterligare placeringsinformation saknas, eftersom plan eller liknande information inte går att koppla till väggarna i fråga.

3.3 Användning av CoClass och information i Solibri

Solibri Model Checker är ett program som kan användas för att sortera och kontrollera BIM – modeller från andra program, exempelvis Revit. Man kan bland annat sortera efter olika "Identity data", vilket är den attribut CoClass – koden har med sig från Revit. Det går även att se hur enskilda objekt ansluter till resten av modellen, vilket kan ge en grund för en teoretisk uppskattning gällande automatisering av CoClass – beteckningar.

3.3.1 Information i Solibri

Det finns ett flertal stora fördelar med att exportera BIM – modeller till Solibri för kontroll. De främsta fördelarna listas här i punktform:

- Informationen i modellen är mer lättillgänglig i Solibri. Solibri skapar ingen ny information, men information som i exempelvis Revit hade varit svårtillgänglig eller gömd går enkelt att hitta och bearbeta i Solibri.
- Det är enkelt att med hjälp av olika verktyg välja vad som visas i 3D – modellen. Om isolerade områden eller objekt behöver studeras går det enkelt att släcka eller gömma omkringliggande objekt för att isolera objektet i fråga. Man kan även spara den redigerade vyn för framtida bruk.
- Solibri kan sortera utefter information som finns i BIM - modellen. Exempelvis går det att välja att bara visa alla dörrar som ligger på plan 2 i en byggnad. Vidare kan man välja att bara visa en typ av dörrar, eller en specifik dörr – förutsatt att BIM – modellen innehåller denna information. Detta verktyg är värdefullt vid bland annat mängdning. Det är även här som CoClass kan komma att spela en stor roll, vilket diskuteras mer i kapitel 3.3.2.

3.3.2 Interaktion mellan Solibri och CoClass

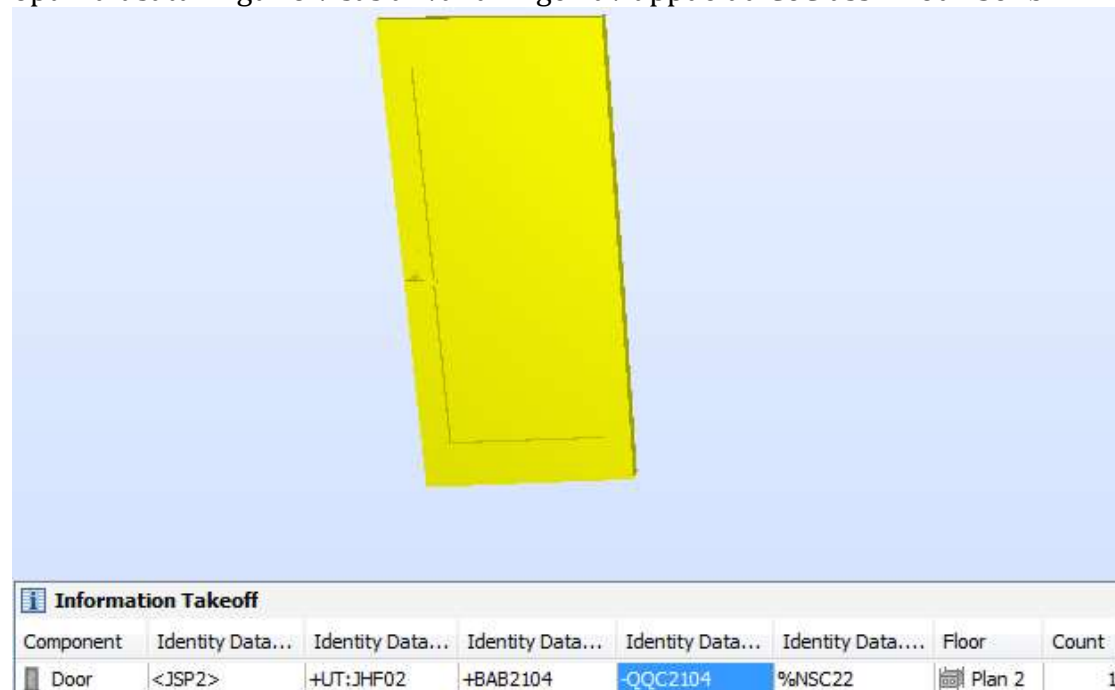
Solibri använder sig av ett stort antal verktyg som kan använda sig av informationen som finns i en BIM – modell. Detta gäller både BIM – modellens egen information, och information som läggs till manuellt. Det finns alltså stora möjligheter att använda CoClass för att förenkla de kontroller som görs i Solibri. Nedan listas och förklaras de främsta interaktionerna mellan CoClass och Solibri:

- Ett moment där korrekt implementering av CoClass i ett projekt kan bidra till stora förenklingar av kontroller i Solibri är mängdning. Detta gäller för alla stadier under ett projekt, men märks främst i senare skeden, eftersom basinformationen i en BIM – modell ofta är nog för en första mängdning. Om exempelvis dörrtyp ännu inte är bestämt för ett projekt kan det i allmänhet räcka med att veta om det är en ytter- eller innerdörr för att kunna göra en grov uppskattning. Om man däremot har specificerat exakta dörrmodeller och lagt in CoClass – kod för detta i modellen går det smidigt att sortera bland och räkna de olika typerna genom att sortera via den skrivna koden. Kort sagt ger en implementering av CoClass en möjlighet att använda Solibri för att göra exakta mängdningar smidigare och snabbare, eftersom den exakta informationen för varje objekt finns i modellen och kan användas vid sortering och urskiljning av specifika typer av objekt.
- Samma princip som för mängdning gäller även vid granskning av projekt. Större sorteringsmöjligheter med hjälp av CoClass – kod bidrar till bättre manövreringsmöjligheter i modellen. Det gör det även enklare att hitta de objekt eller komponenter man vill granska, i och med att det helt enkelt går att leta upp koden som hör till.
- Slutligen underlättar en välgjord BIM – modell när det gäller logistik i ett byggprojekt. Även här kan Solibri dra nytta av att CoClass – kod finns tillgänglig i modellen. Den främsta fördelen har visat sig vara att det snabbt och enkelt går att hitta exakt information gällande objekt i modellen. Det sparar mycket tid och resurser när det gäller att planera leveranser och beställningar, och kan dessutom användas för att optimera projektets tidplan.

3.3.3 Uppdelning av kod i Solibri

I Appendix A1: Referensbeteckningar och exempel (Svensk Byggtjänst, 2016b) beskrivs koden som en enda lång rad med bokstäver, siffror och tecken. På grund av detta skrevs även all kod som användes i kapitel 3.1 i en löpande rad. Under arbetet i Solibri observerades dock vissa problem med det formatet av kod. Det största problemet är att de aktiviteter som finns beskrivna i kapitel 3.3.2 begränsas kraftigt av att varje objekt enbart är utrustad med en unik CoClass – kod.

Det mest logiska alternativet som framkom i undersökningen är att dela upp koden och skapa en parameter för varje del av koden. Detta tillvägagångssätt förenklar och förbättrar användningen av CoClass i Solibri. En parameter för varje del av koden tillåter Solibri att sortera enligt CoClass – kod på ett mer optimalt sätt. I figur 6 visas användningen av uppdelad CoClass – kod i Solibri.



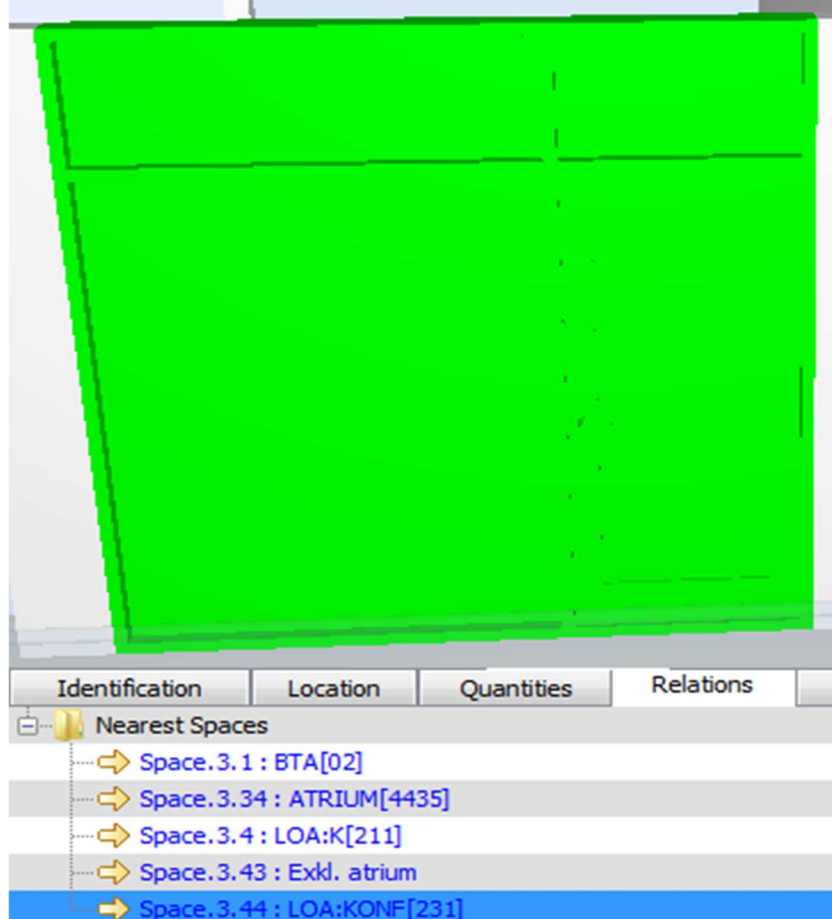
Figur 6: Uppdelad CoClass - kod i Solibri. (<JSP2> - Toppnoden. +UT:JHF02 – Utrymme på plan 2. +BAB2104 – Mötesrum 2104. -QQC2104 – Dörr 2104. %NSC22 – Typ – ID för dörr.)

3.4 Automatisering av kod

I detta kapitel undersöks den teoretiska möjligheten att i framtiden kunna skapa ett system som automatiskt skapar korrekt CoClass – kod till objekt och utrymmen. Figur 6 visar ett exempel på en CoClass – kod från undersökningarna i kapitel 3 i uppdelad form. Varje del av koden presenteras nedan, med tankar om möjlighet för automatisering. Alla slutsatser som dras i detta kapitel gäller enbart i teorin.

- <JSP2> - Toppnoden är den första delen i koden. Toppnoden tillhör alla objekt, utrymmen och annat som finns i projektet i fråga. På grund av detta är det relativt enkelt att skapa en parameter av typen *type* som tilldelas allt som skapas i projektmodellen. Denna del kan alltså, i teorin, enkelt automatiseras.

- +UT:JHF02 – Denna del av koden visar att dörren i figur 6 tillhör ett utrymme på plan 2 i projektet. Om det förutsätts att projektet i fråga har försetts med korrekta planangivelser innehåller dörren i figur 6 den informationen. Det innebär att det är möjligt att skapa en parameter av typen *type* som gäller alla objekt, utrymmen och liknande som befinner sig på plan 2. Detta skulle sedan, i teorin, kunna automatiseras.



Figur 7: Relationer i Solibri.

- +BAB2104 – Denna del av koden visar att dörren ligger i ett mötesrum med nummer 2104. Här upptäcktes vissa problem när det gäller automatisering. Dörren i figuren innehåller information om vart den finns i förhållande till andra objekt, och vilka utrymmen som den gränsar till. En granskning av en vägg i Solibri visar dock tydligt att dörren inte kan avgöra vilket av de angränsande utrymmen som den ska tillhöra. I figur 7 kan man se att Solibri hittar fem utrymmen som gränsar till den belysta väggen. I teorin skulle det vara möjligt att automatiskt välja ett av dessa utrymmen till koden, men det finns inga garantier att rätt utrymme väljs. Ett annat problem uppstår när det gäller tillhörande nummer, i detta fall 2104. I teorin är det fullt möjligt att tänka sig att en nummergegenerator helt enkelt lägger på nummer på exempelvis alla dörrar i projektet, och då startar från ett och jobbar sig uppåt. Detta kan dock leda till en rörig numrering, vilket är negativt då det kan leda till förvirring. Denna parameter verkar alltså, i teorin, svår att automatisera på ett bra sätt.

- -QQC2104 – Denna del av koden betyder att objektet är en dörr med nummer 2104. Dörren i modellen innehåller information som säger att det är en dörr, så på grund av det är det möjligt att skapa en parameter av typen *type* som koppla till alla dörrar i projektet. Däremot uppstår samma problem med nummerordningen som i punkten ovan. Det går alltså teoretiskt sett att automatiskt få med QQC, men det tillhörande numret är mer problematiskt.
- %NSC22 – Denna del av koden är en typ-ID, vilket innebär svårigheter för automatisering. En typ-ID för ett objekt kan beteckna en projektspecifik typ av, i detta fall, en dörr. Det skulle även kunna vara den beteckning som tillverkaren sätter på dörren, och ett antal andra saker. Eftersom objekt i en BIM – modell i de allra flesta fall inte innehåller så pass specifik information medför det att automatisering av typ-ID, i teorin, verkar vara svårt, om inte omöjligt. Dock sätts denna kod på typ objekt dvs. om alla dörrar är av samma typ i byggnaden räcker det att namnge typ-ID på ett ställe i BIM-systemet (Revit).

4 Diskussion

Detta kapitel består av en diskussion gällande oklarheter och slutsatser som uppkom under arbetets gång.

4.1 Uppdelning av kod

Att dela upp CoClass – koden i mindre bitar har fördelar för arbete i Solibri, vilket diskuteras i kapitel 3.3.3, men det är inte den enda fördelen. Det har även visat sig ge fördelar gällande förvaltning. Som diskuteras i kapitel 2.3.1.4 kan en välgjord BIM – modell vara till stor hjälp under en byggnads förvaltningsstadie. Implementering av CoClass i modellen ger en sådan förvaltningsmodell än mer användbar. En uppdelning av CoClass – koden ger en bättre överblick, och bidrar till att det går enklare att hitta den specifika del av kod som eftersöks. Om det exempelvis är koden för en dörr som ska ändras skulle parametern för koden kunna heta *CoClass_Door*. Hade dörrkoden istället varit en del av en lång kodsträng är det inte lika självklart vilken del av koden som representerar dörren i fråga, framför allt om personen som utför ändringen är oerfaren i användande av CoClass. Undersökningen har alltså visat att en uppdelning av koden kan spara både tid och besvär vid förvaltningsarbete.

4.2 Automatisering av CoClass - kod

Automatisering av kod enligt CoClass är något som borde eftersträvas, åtminstone till viss del. Ett problem som uppkom under studien är att implementering av CoClass i en BIM – modell är tidskrävande. Det innebär även att kostnaden för en sådan implementering blir förhållandevis stor. Om det var möjligt att automatiskt generera CoClass – kod skulle det spara både tid och pengar, och därmed förenkla implementering av CoClass i byggbranschen.

Den teoretiska undersökningen i kapitel 3.4 visar dock på att vissa typer av kod är omöjlig att generera. Den visar även på att vissa delar av koden teoretiskt sätt går att generera. Går det att lösa den delen skulle arbete med CoClass gå snabbare och smidigare. Det skulle även bli enklare och mer attraktivt för företag i branschen att aktivt börja använda CoClass i sina projekt.

5 Slutsatser

- CoClass bibliotek och tabeller är väl utrustade och når upp till CoClass mål om att vara flexibelt och anpassningsbart.
- Uppdelning av kod är alltid att föredra när man för in CoClass – kod i en BIM – modell.
- Automatisering av CoClass – kod är teoretiskt sätt möjlig för de delar som faller under kategorin "*type*", det vill säga de delar av koden som inte är unika. Dock "ärver" de information från samma typ vilket gör att det oftast räcker att lägga informationen på ett typ-objekt.
- En lösning för att möjliggöra automatisering av CoClass – kod skulle innebära en stor förenkling för användningen av CoClass, och är därför något som bör eftersträvas.

6 Framtida studier

CoClass är fortfarande i ett tidigt skede, och det finns ett flertal relevanta områden som lämpar sig för framtida studier.

Detta examensarbete har fokuserat uteslutande på byggnadsdelar och utrymmen, och har därför inte undersökt hur CoClass fungerar för andra typer av objekt, exempelvis komponenter och installationer. Detta vore en bra idé för en framtida undersökning.

Ett annat intressant ämne vore en mer ingående undersökning av automatisering av CoClass – kod. Undersökningen i detta arbete är rent teoretisk, så en undersökning som går mer på djupet och utför praktiska försök vore av stort intresse.

7 Referenser

Eastman, C. (2011). *BIM Handbook - A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. Hoboken, New Jersey, USA: John Wiley & Sons, Inc.

Hardin, B. (2009). *BIM and Construction Management - Proven Tools, Methods, and Workflows*. Indianapolis, Indiana, USA: Wiley Publishing, Inc.

Skovgaard, M. (2013). *Cuneco – centre for productivity in construction*. Hämtad 2017-06-04, från: <http://cuneco.dk/english>

Svensk Byggtjänst. (2016b). *Appendix A1: Referensbeteckningar och exempel*. Hämtad från: <https://static.byggtjanst.se/coclass/pdf/Appendix-A1-Exempel-referensbeteckningar-v1.1-20161122.pdf>

Svensk Byggtjänst. (2017a). *BSAB*. Hämtad 2017-05-20, från: <https://byggtjanst.se/tjanster/bsab/>

Svensk Byggtjänst. (2016a). *CoClass – Nya generationen BSAB Klassifikation och tillämpning: En gemensam informationsstruktur genom hela livscykeln för all byggd miljö*. Hämtad från: <https://static.byggtjanst.se/coclass/pdf/Slutdokumentation-CoClass-v1.2-20161026.pdf>

Svensk Byggtjänst. (2017b). *Om CoClass – nya generationen BSAB*. Hämtad 2017-05-25, från: <https://coclass.byggtjanst.se/sv/om#om-coclass>

Svensk Byggtjänst. (2017c). *Tabeller*. Hämtad 2017-05-14, från: <https://coclass.byggtjanst.se/sv>