

Projektering av passersystem

En studie i användarstöd vid offertarbete
för ARX passersystem

Leonard Bachner



UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Calculating ARX access control system proposals

Leonard Bachner

To have a tool that guides you when you work with a complex system can be helpful since the complexity of the work can be reduced. In this thesis I explore the possibilities to use a helper application whilst planning installations of access control systems. The purpose of the application is to provide support during the planning and at the same time make the user aware of the limitations that exist in the access control system ARX Access. By combining early user input and the theory of formativity, the result became an application written in Java.

Handledare: Peter Bergström
Ämnesgranskare: Olle Ericsson
Examinator: Elisabet Andrésdóttir
ISSN: 1650-8319, UPTEC STS08 030

Populärvetenskaplig beskrivning

Att skapa en offert för ett system med många inbördes beroende förhållanden och begränsningar är inte bara tidkrävande utan innebär även en viss risk för fel. När ett passersystem ska projekteras måste en rad faktorer tas hänsyn till såsom komponentkompatibilitet, avstånd mellan komponenter, integration till andra system etc.

Jag har med utgångspunkt av intervjuer hos installatörer av ARX passersystem undersökt vilka svårigheter som finns vid beräkning, projektering och planering av ARX passersystem.

Syftet med detta examensarbete är att underlätta projekteringen av ett passersystem. Grunden i examensarbetet är en applikation skriven i Java. Basen i applikationen är ett gränssnitt där användaren kontinuerligt görs medveten om passersystemets begränsningar och på så sätt både lär sig om systemet samt minskar risken för att planera fel. Detta gränssnitt är utformat som ett träd. Trädets rot är själva passersystemet. Roten är den enda delen av passersystem som finns från början, sedan får användaren via kontextmenyer bygga ut passersystemet. När användaren står på en viss nivå i trädet och tar fram kontextmenyn visas möjliga byggstenar för just den nivån. Därmed får användaren lättare en överblick av vad som fungerar ihop med vad, och lär sig därmed passersystemet samtidigt som en offert för passersystemet räknas fram. En utav de saker som upplevs som svårast vid projektering av ARX passersystem är licenshanteringen. När användaren specificerat den hårdvara som ska ingå i passersystemet kommer därför applikationen ge ett förslag på licenser, som dels baserar sig på vilken typ av hårdvara som valts, samt hur många dörrar angetts. När önskade komponenter valts och licenserna specificerats, räknar applikationen ut totalkostnaden för produkterna samt genererar en artikelnummer lista över alla komponenter som ingår.

Innehållsförteckning

1	Inledning	3
1.1	Syfte	3
1.2	Metod	4
1.2.1	Användarintervjuer	4
1.2.2	Informella intervjuer	5
1.2.3	Min roll, mitt verktyg?	5
1.3	Rapportens disposition	5
2	Bakgrund	7
2.1	R-Design	8
3	Teori	10
3.1	Ett formativt angreppssätt på passersystem	10
3.2	Användarcentrerad systemdesign	10
3.2.1	Design för användbarhet	11
3.3	Java	11
4	Resultat	12
4.1	Intervjuer	12
4.2	ARX Projekteringsverktyg	13
4.2.1	Struktur	13
4.2.2	Flik 1	14
4.2.3	Datatypen Hardware	14
4.2.4	Lägga till en komponent	15
4.2.5	Ta bort en komponent	16
4.2.6	Flytta en komponent	16
4.2.7	Dörrguiden	17
4.2.8	Lägga till en dörrtyp	18
4.2.9	Placera ut dörrtyper i trädet	19
4.2.10	Flik 2	20
4.2.11	Programpaket	20
4.2.12	Dörrlicenspaket	21
4.2.13	Röda alternativ	21
4.2.14	Klientlicenser	22
4.2.15	Larmlicenser	22
4.2.16	Flik 3	23
4.2.17	Datatypen tableData	23
4.2.18	Lägga till och ta bort en komponent	23
4.2.19	Systemets kostnad	24
4.2.20	Använda priser	24
4.2.21	Exempel	25
5	Slutsatser	26
5.1	Kalkylering och Projektering	26
5.1.1	Användartyp 1	26
5.1.2	Användartyp 2	26
5.1.3	Användartyp 3	27
5.1.4	Inriktning på en grupp	27

6	Referenser	28
7	Appendix A – Användarmanual	29

1 Inledning

Jag fick min första riktiga kontakt med passersystem när jag började arbeta på ASSA år 2006. Som ny på avdelningen elektroniska dörrlås, som arbetar passersystem, upplevde jag själv de svårigheter som det innebär att förklara hur passersystem fungerar. Innan jag började jobba på ASSA sträckte sig min erfarenhet av passersystem till just användningen av dem, exempelvis vid tillträde till skolan efter klockan fem och vid tillträde till datorsalar. Jag hade heller inte tidigare reflekterat över hur en kortläsare kunde ta ett beslut om huruvida mitt kort var giltigt eller att passersystem och larmsystem exempelvis kan vara kopplade till varandra, för mig var det ingen skillnad på dessa två system.

Min första uppgift på ASSA var att gå igenom en inkorg för e-post, innehållande gammal korrespondens mellan supportmedarbetare och kunder. För varje brev skulle jag avgöra om brevets innehåll kunde vara relevant för framtida supportärenden och i sådana fall skriva om innehållet till en fråga och ett svar samt kategorisera problemet efter problemområde. I början var det naturligtvis näst intill omöjligt att avgöra relevansen av någonting jag knappt förstod. Det blev heller inte lättare av att ett problem som uppträdde i en viss version av ett passersystem kanske inte alls var ett problem i en annan version eller att passersystemen utvecklades av och för personer som förutsätts ha jobbat med passersystem ett tiotal år¹.

Undan för undan, i takt med att jag fick nya arbetsuppgifter började dock förståelsen växa över hur passersystem fungerar, dels hur läsare är kopplade till varandra men framförallt hur det fungerar rent logiskt, d.v.s. vad som avgör om en person medges tillträde genom en dörr eller inte.

Detta var hur jag för första gången kom i kontakt med passersystem ”på riktigt”. Rimligtvis går det att dra paralleller med hur inläring av andra system inom andra områden. Nästan alla personer kommer idag i kontakt med ett eller flera system som han eller hon förväntas kunna arbeta med eller åtminstone känna till.

Erfarenheterna från när jag började jobba på ASSA är något som jag haft i åtanke undertiden jag arbetat med examensarbetet som denna rapport är en del utav. I de fall jag stött på problem har jag försökt gå tillbaka ca 1 år i tiden för att minnas vad jag själv tyckte var svårbegripligt för att på så sätt kunna fokusera på att förklara dessa delar.

1.1 Syfte

Syftet med detta examensarbete är tvådelat. Det första är att *skapa ett verktyg som underlättar planeringen av ARX passersystem genom att minimera riskerna för fel samt att hjälpa användaren att förenkla projekteringsprocessen av det passersystem som efterfrågas*. Det kan lätt vara hänt att små detaljer i planeringen blir fel, fel som senare kan visa sig göra passersystemsinstallationen omöjlig att driftsätta. För vissa dörrtyper

¹ Detta manifesterades bland annat genom att en ingång på en produkt med tre ingångar kallades ingång 8. Att den heter ingång 8 beror på att det i en tidigare produkt fanns åtta ingångar och den åttonde ingången i den gamla produkten har samma funktion som ingången på den nya produkten som kallas ingång 8.

krävs exempelvis en dörrkontrollenhet av typ DAC-430, det är då viktigt att passersystemet inte byggs ut med dörrkontrollenhet DAC-420.

Det andra syftet är att *skapa ett pedagogiskt verktyg, som förklarar hur hårdvaran i ett passersystem hänger ihop men även hur valet av hårdvara påverkar vilka licenser som behövs för att passersystemet skall fungera*. Att känna till hur passersystemets hårdvara hänger ihop är viktigt för att kunna förstå passersystemets begränsningar. Att känna till hur Passersystemets hårdvara hänger ihop med licenserna är viktigt eftersom detta fungerar på ett unikt sätt i ARX passersystem jämfört med andra passersystem och upplevs ibland som svårt².

Ett verktyg som uppfyller dessa två syften skapar förhoppningsvis ett hjälpmedel som möjliggör för personer som jobbar med passersystem att inte bara snabbare kunna sätta sig in i hur passersystemet fungerar utan även snabbare kunna skapa en offert som med större sannolikhet är korrekt.

Helt konkret innebär detta att min uppgift var att skapa ett hjälpmedel som såg till att hårdvara, mjukvarufunktioner och licenser överensstämmer med varandra. Verktygets slutprodukt skall vara en sammanfattning på vad som ingår i installationen med tillhörande artikelnummer, E-nummer och pris

Så som uppgiften utformades ingick även att verktyget ska kunna ta hänsyn till en 75 procents regel där kapaciteten på passersystemet som mest är 75 %. Detta för att möjliggöra utbyggnad med de kvarvarande 25 procenten, vilket är en i dagsläget vanligt tillämpad metod. Det ska finnas två sätt att bygga ut passersystemet på, antingen genom ett guide läge med frågor där man sedan själv ska kunna gå in och ändra antalet dörrar eller ett manuellt läge där man bygger upp hela systemet självt.

1.2 Metod

För att uppnå mitt syfte har jag först av allt behövt ta reda på vad som upplevs som svårt när ett passersystem i allmänhet och ARX passersystem i synnerhet ska projekteras, för att sedan skapa det verktyg som förhoppningsvis ska kunna överbygga just dessa svårigheter.

1.2.1 Användarintervjuer

För att ta reda på vad som upplevs som svårt vid projektering av passersystem genomförde jag ett antal användarintervjuer. I detta ändamål kontaktade jag tre företag som jobbar med projektering av ARX passersystem. På dessa företag genomförde jag totalt sex stycken intervjuer. Tre stycken intervjuer genomfördes i början av projektet för att bilda mig en uppfattning om projekteringsprocessen. De frågor som intervjuerna kretsade kring var:

- I vilken omfattning de arbetade med just *ARX* passersystem.
- Hur de arbetade med ARX passersystem.
- Vad som var svårast under arbetet med att projektera ARX passersystem och vad en projekteringsapplikation kunna underlätta.

² Intervju 17/09/08

När jag genomfört de initiala intervjuerna påbörjades arbetet med att göra en prototyp på applikationen. Efter att ha testat en del möjliga gränssnitt och bestämt det slutgiltiga utseendet påbörjade jag utvecklingsarbetet. För att sedan kontrollera att jag var på rätt väg och för att om möjligt kunna korrigera det som var fel samt för att kunna förslå förbättringar för vidareutveckling genomförde jag efter vecka 15 de sista intervjuerna.

1.2.2 Informella intervjuer

Förutom de intervjuer jag genomfört och som diskuteras ovan har jag haft dagliga samtal med medarbetare på ASSA, om passersystemens funktioner, vilket informationsbehov som finns vid projektering av passersystem samt hur verktyget som denna rapport bygger på skulle kunna se ut. Dessa samtal har för detta verktygs utformande, varit ovärderliga. Ur ett vetenskapligt perspektiv skapar de dock ett transparensproblem. Dessa samtal har, på grund utav dess informella natur, varken skrivits ner eller på annat sätt diarieförts, delvis för att jag vid tidpunkten då samtalet ägde rum inte visste att jag skulle skriva mitt examensarbete om just detta ämne och delvis eftersom de skett spontant när papper och penna inte funnit till hands. En fara med detta är naturligtvis att minnen förändras och förvanskas med tiden.³

1.2.3 Min roll, mitt verktyg?

När ett verktyg i likhet med det detta, skapas, är ett återkommande problem för vem verktyget skapas. Irene Molina skriver:

*”Det är också nödvändigt att vara medveten om att det i samhällsvetenskaplig forskning inte existerar en enda sanning. Vilken sanning som kommer fram, beror i hög grad på vem som berättar den.”*⁴

Draget till sin spets, menar Molina att det finns en risk som innebär att jag skapat ett verktyg som är utformat speciellt för mig eftersom det är jag som tolkar de jag intervjuar och det är mina egna erfarenheter samt minnen från den tid då jag just börjat som ligger till grund för verktyget. Denna risk ökar ännu mer med tanke på att jag baserat utformningen av verktyget på odokumenterade samtal som ibland skett för över ett år sedan. För att i någon mån kompensera för detta, har jag försökt att lägga så stor vikt som möjligt vid de 6 intervjuer jag genomförde ute hos de tilltänkta användarna av detta verktyg.

1.3 Rapportens disposition

En introduktion till passersystem i allmänhet och ARX passersystem i synnerhet finns i kapitel 2. De komponenter som kan ingå i ett passersystem är inte alldeles självförklarande, därför förklarar jag de vanligaste komponenterna samt dess funktion här. I detta kapitel visar jag också en på en liknande ansats till lösning på hur ett projekteringsverktyg skulle kunna se ut. Detta verktyg är dock utformat för ett annat passersystem varför jag egentligen inte kan avgöra verktygets kvaliteter mer än på en principiell nivå. Kapitel 3 tar upp de teoretiska utgångspunkterna som jag använt mig av. I själva verket bygger detta examensarbete till stor del på ett experimentellt

³ Bergquist, Johan. 1996. Vem kan man lita på?. Om minnets exakthet och motståndskraft mot påverkan

⁴ Molina, Irene. 1997. s. 17. Stadens rasifiering. Etnisk bostadssegregation i folkhemmet.

tillvägagångssätt som till stor del gått ut på att pröva mig fram. Detta beror till stor del på att det varit svårt att bedöma tidsåtgången. En teori som trots allt väglett mig genom hela programmet handlar om att inte begränsa användaren till det som betraktas som "den bästa lösningen". I Kapitel 4 beskriver jag det som blev detta examensarbets slutprodukt, denna rapport undanräknad. Jag har valt att inte beskriva programmet metod för metod eftersom det förmodligen, av läsaren, skulle uppfattas som osammanhängande och ointressant. Samtidigt, är det motiverat att beskriva de funktioner som finns i programmet mer detaljerat än ur ett användarperspektiv. Därför redogör jag på en principiell nivå jag vad som händer när verktygets komponenter manipuleras. I det sista kapitlet, kapitel 5, diskuterar jag de val som gjorts under arbetets gång samt möjliga vägar att vidareutveckla verktyget. Som appendix har jag lagt användarmanualen till verktyget, denna kan vara bra att ha till hands när och om projekteringsverktyget används.

2 Bakgrund

Passersystem finns idag i en rad olika tappningar och utföranden och många utav oss möter dagligen ett antal utav dessa, på väg till och från arbete och skola. Huset vi bor i har ofta någon form av passersystem, vare sig det är ett kodlås på dörren eller en läsare som läser kodbärare. När vi tar tunnelbanan eller bussen möts vi av ett passersystem som kontrollerar om vi har ett giltigt färdbevis. När vi åker mellan Uppsala och Stockholm möts vi av ett passersystem i form av en elektronisk biljettförsäljare. De flesta arbetsplatser har också någon orm av passersystem installerat.

Var och ett utav dessa passersystem har olika syften. Passersystemet i huset där vi bor har till syfte att förhindra ej välkomna personer inträde och då räcker det eventuellt med en kod som är känd för alla som bor i huset samt deras vänner och bekanta.

Passersystemet i tunnelbanan och bussarna syftar till att kontrollera att vi har en giltig färdbiljett. Passersystemet på bussar och tåg mellan Stockholm och Uppsala tar betalt för våra resor. Passersystemet på jobbet eller skolan syftar även det ofta till att förhindra obehöriga från att inträda, beroende på arbetsplats kan dock kontrollerna vara rigorösare här än till bostaden, så att genom exempelvis vändkors eller slussar förhindra att en person släpper in flera. Det passersystem som utvecklats av ASSA och som detta examensarbete kretsar kring, ARX Passersystem används framförallt som passersystem i byggnader och då framförallt i offentliga samt kontorslokaler.

ARX passersystem bygger på en central server, där databasen med all information om kort, personer, dörrar, tillträdesregler etcetera finns lagrade. ARX passersystem konfigureras sedan via klienter. I klienterna sköts både den dagliga driften, såsom att lägga upp nya personer och kort, men även konfigurationen när nya installationer görs här, exempelvis när ytterligare dörrmiljöer installeras. En klient i detta avseende är en dator med ARX klientprogramvara installerat.

ARX server är sedan kopplad till ett antal centralenheter. Kopplingen mellan ARX server och Centralenheterna sker vanligtvis via Ethernet. Centralenheterna utgör stommen i ARX passersystem och fungerar även utan kontinuerlig kontakt med ARX server, det är här besluten tas om en dörr ska öppnas eller inte. Till ARX passersystem finns det två olika typer av Centralenheter, Centralenhet 9016 för dörrmiljöer och Centralenhet 9017 för porttelefonmiljöer.

I Centralenheterna kan totalt fyra loopkort monteras. Det första loopkortet, ibland kallat basloopkort, följer alltid med centralenheten och skiljer sig en aning från efterföljande loopkort som monteras i centralenheten. Det finns två typer av loopkort som använd för att bygga ut centralenheten, ett loopkort där ytterligare porttelefonmiljöer kan monteras och ett där dörrmiljöer kan monteras. På loopkorten för dörrmiljöer kan totalt fyra dörrmiljöer monteras. På loopkorten för porttelefoner kan totalt två porttelefonmiljöer monteras.

Dörrmiljöerna kan se ut på flera olika sätt. Den mest avancerade dörrmiljön är den som styrs av en dörrkontrollenhet, hit kan både inside- och utsideläsare kopplas, men även motorlås, öppnarknappar m.m. Om ett tillkopplingskort sätts in dörrkontrollenheten ökar konfigurationsmöjligheterna ytterligare, genom att utnyttja de reläer som finns på tillkopplingskortet kan dörrmiljön i princip komma att se ut hur som helst. De läsare

som kopplas in kan t.ex. vara beröringsfria läsare eller magnetkortsläsare. Dörrmiljöerna som saknar dörrkontrollenhet, inklusive porttelefonmiljön, är i regel inte lika flexibla men består ofta av minst en läsare på utsidan av dörren samt ofta en öppnarknapp på insidan. Det maximala avståndet mellan en dörrmiljö och en centralenhet varierar på mellan 300 och 500 meter.

2.1 R-Design

En tidigare ansats för att förenkla processen med att planera en installation av ett liknande passersystem har gjorts. R-design är ett program skrivet i Excel där användaren, utifrån ett antal på förhand definierade dörrmiljötyper, specificerar hur många av varje dörrtyp som ska ingå i installationen.

R-Design Ver 1.4 , © Copyright RCO Security AB , 2003-02-10

ANTAL KORT

0 TAGGAR

LÄSNINGSTEKNIK

MAGNET PROX DALLAS SMART

FÄRG PÅ LÄSARE

VIT GRÅ

STRATEGI UNDERCENTRALER

OPT. 8 dörrar 4 dörrar

STRATEGI DÖRRMILJÖ

IO opt DB & IO DB & Lite bara DB

DÖRRAR - LARM-STYRANDE / FÖRBIKOPPL. LÄSARE

varav UTPASSERING med

VRED + KNAPP + UTLÄS = Summa dörrar

0 + 0 + 0 = 0

Varav utomhus: 0

DÖRRAR - LÄSARE MED KNAPPSÄTS

varav UTPASSERING med

VRED + KNAPP + UTLÄS = Summa dörrar

0 + 0 + 0 = 0

Varav utomhus: 0

DÖRRAR - LÄSARE utan knappsats

varav UTPASSERING med

VRED + KNAPP + UTLÄS = Summa dörrar

0 + 0 + 0 = 0

Varav utomhus: 0

VÄLJ VAD SOM SKA VISAS I LISTAN

ART. NR E-NR C:A PRISER

STRÖMFÖRSÖRJNING

Ingen Trafo UPS

Hjälp

Visa lista

Bild 1, R-design del 1

När användaren specificerat det antal dörrmiljöer som önskas och klickar på knappen "Visa lista", visas en lista över de komponenter som kan ingå i systemet.

PRODUKTNAMN - MODELL	ARTNR	E-NR	C:A pris
R-CARD WIN MINI. v 4.30	20020022	6360421	515 kr
UC-50/4-2	20080100	6360150	6 080 kr
DB-50 Delningsbox	20080300	6360159	2 370 kr
DB-50 Light delningsbox	20080310	6360162	1 800 kr
PROX-62KG	20043210	6364241	5 350 kr
TOTALT:			16 115 kr

bild 2, R-design del 2.

De problem som R-design har är just att dörrmiljöerna är fördefinierade och inte synliga. Om en viss typ av läsare valts, syns inte resten av dörrmiljön. När jag väl valt att visa listan över komponenter kan jag heller inte gå tillbaka och förändra något i specifikationen. En utav de begränsningar som finns i passersystem är att en centralenhet inte kan ligga hur långt ifrån sina dörrcentraler som helst. I R-design tas ingen hänsyn till denna begränsning. Det går heller inte att kontrollera att rätt antal centralenheter är inkluderade. I och med att det endast presenteras en lista saknar programmet en pedagogisk dimension där användaren kan se hur passersystemet är uppbyggt och på så sätt förstår varför uppsättningen av komponenter som får systemet att fungera, ser ut som den gör.

3 Teori

3.1 Ett formativt angreppssätt på passersystem

Kim Vicente diskuterar i boken *Cognitive Work Analysis*⁵ koncepten kring en formativ arbetsanalys. Begreppet har uppkommit kring studier av realtidsystem⁶ men jag tror att de poänger Vicente drar även går att överföra på andra situationer, som alternativ till ett normativt synsätt.

Det normativa synsättet utgår från att de som designat systemet bäst vet vad det kan användas till vilket får till följd att systemet endast ger användarna stöd i det arbete som systemdesignern haft för avsikt att systemet ska hantera.⁷

Ett normativt angreppssätt för en applikation som skall hjälpa användaren att bygga upp en passersystemsinstallation skulle exempelvis kunna be användaren att ange hur många dörrmiljöer som skall skapas och löser sedan problemet vilka komponenter som skall ingå. I motsats till det normativa angreppssättet skulle det formativa snarare presentera en arbetsyta som endast begränsas av vad som är, de facto, omöjligt. Den formativa applikationen överlåter sedan till användaren att uppfylla sina mål, där det inte utesluts att det finns fler än ett sätt att nå samma mål.⁸

Användarens uppgift blir i det här fallet att själv skapa passersystemet, det skapar ett visst merarbete men ökar samtidigt flexibiliteten över vilka mål passersystemet kan uppfylla. Samtidigt som passersystemet nu blir flexiblare, läggs också ett ansvar på användaren att systemet fungerar i enlighet med det mål som satts upp för systemet.⁹

Sett till passersystem blir det möjligas gräns dock inte helt enkel att dra. Resonemanget om att antingen fungerar enheterna ihop, eller så gör dom det inte bär fortfarande spår av ett normativt sätt att resonera. En öppnarknapp fungerar inte när den är inkopplad direkt på centralenheten. Det vill säga den fungerar inte i den bemärkelse vi normalt sett syftar på, den kan inte öppna en dörr på detta sätt. Här har jag emellertid valt att dra gränsen för passersystemets begränsningar.

3.2 Användarcentrerad systemdesign

Ett mål med programmet är naturligtvis att det bör vara användarvänligt, jag har därför tagit fasta på ett antal principer för god design som Gulliksen och Göransson presenterar i boken *Användarcentrerad systemdesign*¹⁰. Som titeln avslöjar, argumenterar författarna för att systemets användare bör involveras i designprocessen. Nedan följer

⁵ Vicente, Kim. 1999. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*.

⁶ Vicente, Kim. 1999. s. 4

⁷ Vicente, Kim. 1999. s. 61 ff.

⁸ Vicente, Kim. 1999. s. 123

⁹ Vicente, Kim. 1999. s. 124-130

¹⁰ Gulliksen, Jan & Göransson Bengt. 2002. s. 107. *Användarcentrerad systemdesign*.

några utav de punkter jag försökt följa vid utvecklingen av detta projekteringshjälpmedel.

3.2.1 Design för användbarhet

- Tidigt och kontinuerligt fokus på användarna – Designern måste ta reda på vilka användarna är. Sedan ska man studera dessa användares sätt att uppfatta och se på sina arbetsuppgifter för att på så sätt kunna ta till vara på användarnas styrkor och stödja användarna på de områden där det finns kunskapsluckor. Att dessutom involvera användarna genom deltagande design är viktigt.¹¹
- Iterativ design – När problem upptäcks i användartesterna bör de åtgärdas. Detta betyder att processen måste vara iterativ. Med andra ord måste det finnas en cyklisk process av design, utvärdering och omdesign etc. Empirisk mätning och iterativ design är nödvändig eftersom designern, oavsett skicklighet, inte kan komma fram till den rätta lösningen på en gång.¹²

3.3 Java

Vid val av utvecklingsverktyg för applikationen, valde jag Java. Valet beror dels på grund av Javas flexibilitet. Applikationen kan köras som enskild applikation, inom ramen för en webbläsare eller från en Server.¹³ I valet mellan dessa tre finns naturligtvis en rad för och nackdelar men ytterligare en fördel är att det går relativt smärtfritt att byta mellan dessa plattformar. En annan viktig orsak till att applikationen utvecklades i Java är för att det är det språk jag känner mig bekväm med.

¹¹ Gulliksen, Jan & Göransson Bengt. 2002. s. 107.

¹² Gulliksen, Jan & Göransson Bengt. 2002. s. 108.

¹³ Java. <http://www.sun.com/java/about/index.jsp>

4 Resultat

4.1 Intervjuer

Intervjuerna visade på att de som jobbar med att projektera passersystem, överlag, förefaller ha en god insikt över hur passersystem fungerar och ser ut. I den mån de saknade erfarenhet från ARX passersystem, kände dom till liknande system, så som andra passersystem eller larmsystem. På grund av detta borde ett projekteringsverktyg förutsätta en relativt god kunskap om passersystem i allmänhet.

Intervjuerna tyder också på att de som projekterar ARX passersystem gör detta som en uppgift bland andra, med andra ord förefaller det ovanligt att på heltid projektera ARX passersystem. Därför borde ett projekteringsverktyg fokusera på att hjälpa till med detaljerna samt det som är unikt i passersystemet.

Sett till passersystem, handlar det normativa synsättet om vad en dörrmiljö är och vad en dörrmiljö ska göra. Den normativa dörrmiljön har en läsare på utsidan och en knapp och eller läsare på insidan. Problemet med R-design är just detta normativa tänkande kring vad en dörrmiljö är. Detta skapar inte bara problem när vissa typer av dörrmiljöer som användaren vill skapa inte går att skapa eftersom de inte finns med i programmet, det kan också skapa en osäkerhet kring vilka typer av dörrmiljöer som går att skapa och vilka som inte går att skapa.

Mitt mål med denna applikation var således att i så liten grad det bara var möjligt påverka användaren i hur dörrmiljön skulle och bör se ut. Vill användaren skapa en dörr med tre läsare, där var och en utav dessa läsare använder sig av olika läsarteknologier, så ska det vara tillåtet. De personer som projekterar passersystem har tillräcklig kunskap om vad detta skulle innebära för dörrmiljön. Applikationen behöver därför inte på en konceptuell nivå ge användaren stöd. Vad jag vill göra är att kunna presentera passersystem utan denna begränsande syn, det som sätter gränserna för passersystemet är inte hur dörrmiljöerna bör se ut, utan hur de kan se ut. Återkopplat till Vicente, applikationen ska presentera systemets gränser, sedan ska det stå användaren fritt att lösa sitt mål på valfritt sätt. En dörrmiljö med tre knappar och ingen läsare kan te sig ologisk, men eftersom en dörrmiljö kan se ut så, så ska det också vara möjligt att bygga upp den just så.

Samma resonemang är emellertid också giltigt vad gäller kopplingen mellan passersystemets licenser och de komponenter som specificerats. Ett normativt angreppssätt vore exempelvis att låta programmet lägga till ett dörrlicenspaket som motsvarar det antal dörrmiljöer som specificerats. Det formativa angreppssättet vore att ge användaren det slutgiltiga bestämmandet över dessa val. Det skulle till exempel kunna tänkas vara så att användaren inte tänkt använda alla de dörrmiljöer som specificerats eller så kanske användaren redan har ett antal dörrlicenser och därför vill köpa färre dörrlicenser än dörrmiljöer.

4.2 ARX Projekteringsverktyg

Applikationen består av tre flikar och tanken är att man ska gå igenom programmet från flik 1 via flik 2 och slutligen till flik 3 men det ska även vara möjligt att gå tillbaka för att ändra, ta bort och lägga till komponenter i efterhand.

Applikationens syfte är som nämnts i inledningen, tvådelat. Dels syftar applikationen till att underlätta förståelsen av hur ett ARX passersystem byggs upp (flik 1), vilka komponenter som kan ingå samt hur kopplingen mellan hårdvara och mjukvara ser ut (flik 2). Dels syftar applikationen till att sammanställa en produktlista över alla komponenter som specificerats till passersystemet (flik 3).

Priserna som används i applikationen tas ur filen Prislsta.skv som ligger i samma katalog som applikationen själv. Om filen Prislsta.skv inte ligger i applikationens katalog används 2008 års priser.

4.2.1 Struktur

Programmet består i sin helhet av 40 stycken java filer, paketerade till en .jar fil¹⁴. 11 av dessa filer är själva grunden i programmet. Eftersom varje del av ARX passersystem, från centralenheter till läsare har vissa unika egenskaper, såsom vilka andra komponenter som kan kopplas ihop med just den komponenten, har varje produkt definierats i en egen .java fil. Totalt finns 29 produkter definierade på detta sätt i programmet. I själva verket finns ytterligare ett antal produkter eftersom exempelvis PCR40 vit och PCR40 svart finns definierade i samma fil. I detta fall är det endast färgen, artikelnumret och E-numret som skiljer produkterna åt. De produkter som inte har en egen fil är de som definieras under flik 2, det vill säga mjukvarulicenserna. Detta beror på att dessa inte behövs kopplas ihop på samma sätt och den enda information som behövs om mjukvarulicenserna, ur ett applikationsutvecklingsperspektiv är dess artikelnummer, E.-nummer, namn, beskrivning och pris och dessa finns definierade inuti programmet. För att kunna köra programmet krävs minst java version 1.6, bland annat för att GroupLayout¹⁵ använts som layoutmetod.

¹⁴ En .JAR fil startas på samma sätt som en .EXE fil.

¹⁵ Java. <http://java.sun.com/javase/6/docs/api/javawx/swing/GroupLayout.html>

4.2.2 Flik 1

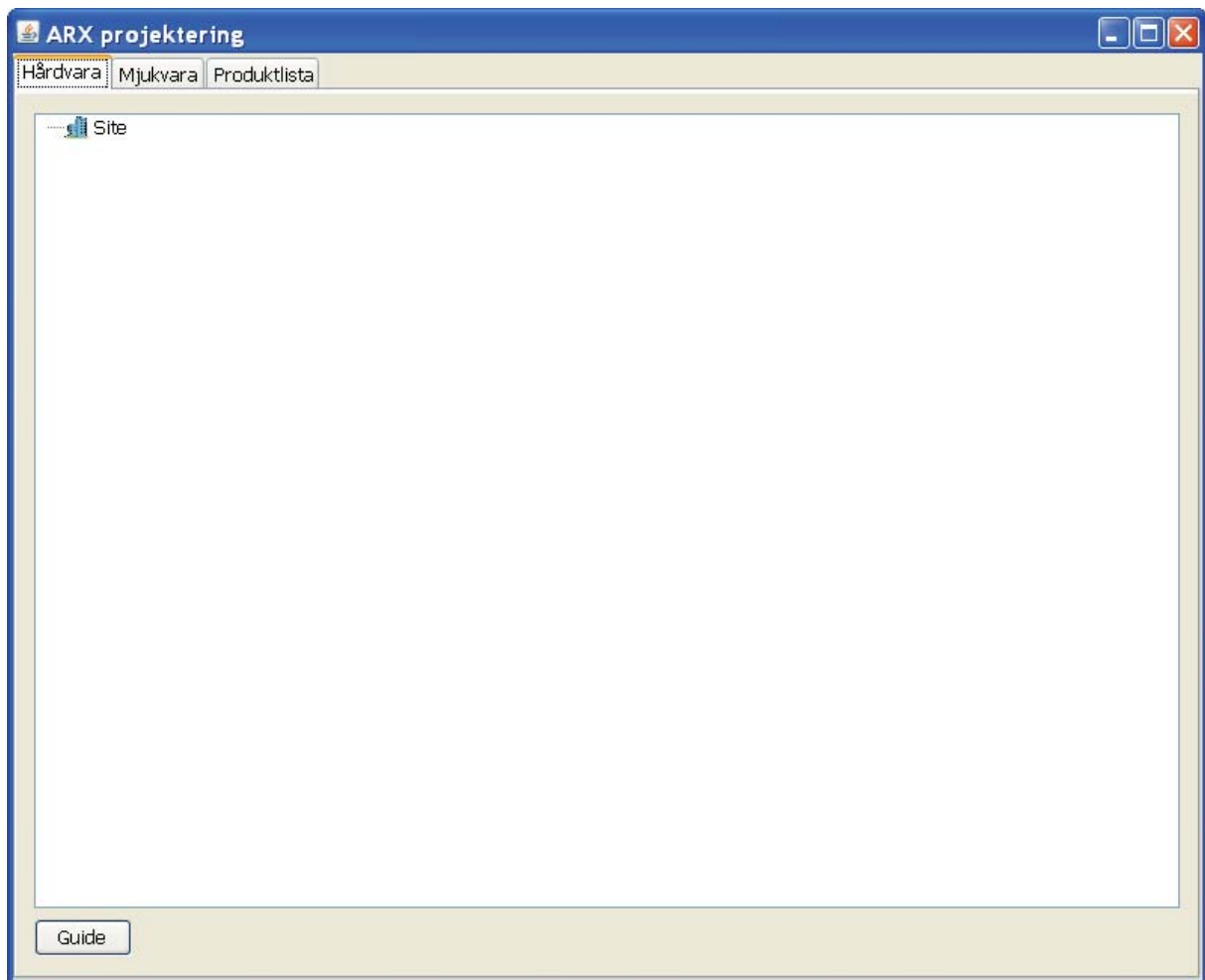


Bild 3, Flik 1.

Flik 1, hårdvara består av ett vitt fält samt en knapp och används för att bygga upp hårdvarukomponenterna till ett ARX passersystem, det är denna flik som först visas när programmet startas. När programmet startas finns endast en ikon i det övre högra hörnet, Site. När fler komponenter läggs till kommer de att placeras i ett träd så att deras inbördes hierarkiska ordning blir tydlig.

4.2.3 Datatypen Hardware

Eftersom de komponenter som läggs till i programmet ska återspegla hur de i verkligheten förhåller sig till varandra, blev det mest pedagogiska perspektivet att placera komponenterna i ett hierarkiskt träd¹⁶. I denna hierarki tydliggörs t.ex. varför två loopkort och sex dörrmiljöer kan användas. Varför det är möjligt att dessa sex dörrmiljöer kopplas in men att det inte går att koppla in 5 dörrmiljöer på ett utav loopkortet och en dörrmiljö på det andra loopkortet. För att avbilda relationerna mellan komponenterna används i flik 1, swingkomponenten Jtree. För att göra det enklare att navigera i trädet, representeras alla enheter av en ikon som liknar det som den i verkligheten ska föreställa¹⁷.

¹⁶ Java. <http://java.sun.com/docs/books/tutorial/uiswing/components/tree.html>

¹⁷ Nygren, Else. 1997. Grafiska användargränssnitt, några tips.
<http://www.it.uu.se/research/hci/publications/papers/60/60.html>

I Java inte finns någon fördefinierad datatyp av trädstruktur. I swingkomponenten Jtree går det att navigera sig vertikalt mellan trädets komponenter, den använda datatypen som komponenterna består av hade alltså egentligen inte behövt vara av trädstruktur (med länkar till tillhörande noder). Nackdelen med att skapa objekt som inte länkar till varandra och placera dom i ett JTree är att man blir beroende av komponenten Jtrees länkar, dessutom är det inte är möjligt att i Jtree specificera vilka komponenter som passar ihop med vilka andra. Istället skapade jag därför datatypen Hardware och varje objekt som skapas i trädet ärver denna datatyp. Med denna datatyp ärvs ett antal metoder som kan användas på objektet samt de variabler som länkar ihop komponenterna med varandra. Varje komponent som ärvt Hardwares egenskaper har information om vilka andra komponenter den hör ihop med men även vilka den är kompatibel med.

Alla komponenter som skapas i trädet har en komponent ovanför sig, denna komponent benämns som förälder. De flesta komponenter som skapas i trädet kan i sin tur ha andra komponenter kopplade till sig, dessa benämns som barn eller extrabarn. Skillnaden mellan de båda är att barnen är de komponenter som vanligtvis läggs till. Extrabarn används om komponenten har några övriga produkter som går att ansluta. Anledningen till att en distinktion behövt göras är på grund av att de går att addera till komponenten enligt olika logik. Centralenhet 9017 har exempelvis följande barn: loopkort 4014 och 9018, totalt kan centralenheten ha upp till fyra loopkort kopplade till sig. Men oberoende av hur många loopkort som faktiskt är inkopplade gåt det dessutom att koppla in ett reläkort 6416 och en bokningspanel 8001, detta är centralenhetens extrabarn.

Alla komponenter som skapas av typen hardware sparas i en Arraylist av typen <Hardware>, kallad Centralerna. Detta innebär att nya objekt som skapas enkelt kan läggas till genom metoden add. Samtidigt innebär det att alla objekt behöver ett unikt id-nummer eftersom dess plats i ArrayList inte är beständig. När ett objekt tas bort i mitten av Arraylist så flyttas positionen på alla objekt som ligger efter det borttagna objektet, ett steg bakåt. Fördelen med denna lösning är att det är lätt att lägga till nya objekt, utan att behöva ta hänsyn till hur många som redan finns, d.v.s. man risker aldrig att överskrida gränsen för listans längd. En annan fördel men denna lösning är att det är enkelt att snabbt få en översikt på vad som finns inlagt, genom att det bara är att gå framåt i listan över installerade komponenter går det snabbt att räkna igenom alla härvarukomponenter som finns i installationen.

4.2.4 Lägga till en komponent

För varje produkt finns en unik metod som lägger till just den produkten, de skillnader som faktiskt finns mellan metoderna beror på logiken kring hur produkten kan läggas till eller inte.

Metoden som lägger till en komponent skapar själva komponenten och placerar den i ArrayListan, och lägger sedan till komponenten, dels till flik 1 i hårdvaruträdet och till flik 3 i kalkylbladet. Det som skiljer metoderna åt är framförallt vart respektive komponent läggs till hos dess förälder, d.v.s. om den räknas som ett normalt barn eller ett extrabarn, såsom reläkorten och bokningspanelen. Vissa metoder är överlagrade, detta beror på att det i vissa fall ska gå att lägga till en komponent till den som för

tillfället är markerad i trädet och i andra fall ska komponenten läggas till någon annan stans. När metoden anropas med en specifik föräldraNod läggs komponenten till denna, när metod anropas utan specifik föräldraNod läggs komponenten till den nod som är markerad.

Principiellt händelseförlopp:

1. Metoden anropas med Id-numret på föräldern.
2. Förälderns plats i ArrayListan letas upp.
3. Kontroll om komponenten kan läggas till föräldern genomförs, dvs. är dom kompatibla med varandra och finns det plats?
4. Ett unikt id skapas till komponenten
5. Komponentens förälder anges idnummret på komponenten.
6. I komponenten anges idnummret på dess förälder.
7. En ny trädnod skapas och komponenten läggs in i denna nod som placeras ut i trädet, antingen på det ställe som angetts eller det ställe som är markerat.
8. Information om att komponenten skapats skickas till kalkylbaldet, flik 3.

4.2.5 Ta bort en komponent

Alla komponenter, utom roten "Site", går att ta bort. När en komponent tas bort kommer även alla eventuellt underliggande komponenter att tas bort. För att ta bort en komponent, högerklickar användaren på komponenten och när menyn visas kommer alltid det sista alternativet vara att ta bort komponenten. Oavsett vilken komponent som tas bort används samma metod.

Principiellt händelseförlopp:

1. Metoden för att ta bort komponenter anropas med idnummret på komponenten som skall tas bort.
2. Komponentens plats i ArrayListan letas upp.
3. Komponentens förälder anropas och den aktuella komponenten tas bort i förälderns lista över vilka barn den har.
4. Kontroll om komponenten som ska tas bort har några barn sker.
5. Om komponenten som ska tas bort i sin tur har barn:
 - a. Id-numret för alla barn, vare sig de är betraktade som barn eller extrabarn kopieras till en Array
 - b. För varje plats i denna Array anropas metoden för att ta bort en komponent
6. Information om att komponenten kommer att tas bort skickas till kalkylbladet, flik 3.
7. Komponentens plats i ArrayListan Centralerna
8. Komponentens plats i swingkomponenten Jtree, flik 1

4.2.6 Flytta en komponent

För att flytta en komponent till en annan plats i trädet, exempelvis för att flytta ett 4014 loopkort från en undercentral till en annan, behöver den endast dras från det ställe där den befinner sig till det ställe som användaren vill placera den på. Användaren får hela tiden återkoppling på var komponenten kan flyttas eftersom pekaren ändrar form beroende på om komponenten kan släppas där den för tillfället befinner sig eller ej. Om flyttningen av någon anledning inte genomfördes, exempelvis när en komponent flyttas

till ett ställe dit den inte kan flyttas, kommer komponenten att ligga kvar på dess ursprungsposition.

Principiellt händelseförlopp:

1. Metoden anropas med noden för komponenten som ska flyttas samt noden för komponentens nya förälder.
2. Komponentens själv, komponentens gamla förälder, komponentens nya förälder laddas in.
3. Komponentens idnummer anges i komponentens nya förälder.
4. Komponentens idnummer tas bort från komponentens gamla förälder.
5. Den nya förälderns idnummer anges i komponenten.
6. Komponentens tas bort från den gamla föräldern i trädet.
7. Komponentens läggs till den nya föräldern i trädet.

4.2.7 Dörrguiden

Namn

Läsare utsida Färg

Läsare insida Färg

Knapp

Motorlås Motorlås

RC64 Reläkort RC64

Antal dörrar

Inga dörrar inlagda

Utnyttja 75% av centralenheternas kapacitet

Minst 0 centralenheter kommer behövas

Bild 4, Dörrguiden.

Knappen på flik 1 leder till att ett nytt fönster öppnas, dörrguiden. Denna guide är egentligen ett alternativt sätt att placera ut passersystems komponenter i trädet under flik 1. Istället för att som i trädet placera ut komponent för komponent skapas i guiden istället dörrtyper. En dörrtyp är en dörrmiljö som kan användas flera gånger. När dörrtyperna är definierade, samt att användaren har specificerat hur många av varje dörrtyp som ska ingå i passersystemet kommer programmet att placera ut dörrmiljöerna på automatiskt skapade centralenheter och loopkort. Användaren kan sedan justera hur dörrmiljöerna ska fördelas över centralenheterna och/eller skapa ytterligare dörrmiljöer direkt i hårdvaruträdet. Att använda dörrguiden kan vara ett effektivt sätt att placera ut komponenter om flera identiska dörrmiljöer ska placeras i hårdvaruträdet. Komponenter som lagts till innan dörrguiden används kommer att tas bort när dörrmiljöerna skapade i dörrguiden placeras ut, detta för att det inte ska skapas ytterligare uppsättningar av dörrmiljöer varje gång dörrguiden startas.

Om vi exempelvis vill ha ett passersystem med fem identiska dörrmiljöer, så räcker det här att ange hur en utav dörrmiljöerna ser ut, d.v.s. dörrtypens utseende, för att sedan ange att fem exemplar ska skapas.

Dörrguiden består av en lista, till höger, som visar alla tillgängliga, skapade dörrmiljöer, till vänster finns det nio parametrar som förändrar den aktuella dörrtypens utseende.

Egenskaperna är:

Namn – namn på dörrtypen, det vill säga det namn som kommer att visas i dörrtypslistan till höger.

Läsare utsida – Detta fält anger vilken läsare som skall sitta på utsidan av dörren. Förutom att ange utsidesläsaren, bestämmer detta fält över om vissa av de andrafälten ska vara tända eller släckta.

- Fältet till höger om Läsare utsida, som har två val, svart eller vitt anger färgen på läsaren. Huruvida detta fält är tänt eller släckt styrs av det val som görs i fältet läsare utsida. Exempelvis finns porttelefon ECP30 endast i grå färg och således går det inte att välja mellan svart eller vitt.
- På samma sätt kommer valet av utsidesläsare styra huruvida en dörrkontrollenhet kommer att skapas till dörrtypen eller inte. Detta innebär i sin tur att de val som bara är relevanta då en dörrkontrollenhet finns tillgängliga kommer att tändas eller släckas beroende på val av utsidesläsare. De val som endast finns tillgängliga med en dörrkontrollenhet är kryssrutorna RC64 och Motorlås samt val av insidesläsare.
- Om DBL34 valts som utsidesläsare kommer alla möjligheter att påverka dörrmiljöns utseende att vara släckta.

Läsare insida – Finns endast för vissa val av utsidesläsare. Beroende på val av insidesläsare kommer fältet där dess färg väljs, att tändas eller släckas enligt samma logik som för läsare utsida.

Knapp – Här anges ett val för att ha en öppnarknapp till dörrmiljön.

Motorlås – Kryssas denna ruta i kommer dörrkontrollenhet DAC430 att användas.

Lämnas denna ruta oikryssad kommer dörrkontrollenhet DAC420 att användas.

RC64 - Kryssas denna ruta i ett reläkort RC64 skapas till på denna dörrtyp.

Antal dörrar – Här anges det antal dörrmiljöer som skall skapas av den aktuella dörrtypen. Giltiga värden är positiva heltal.

4.2.8 Lägga till en dörrtyp

När en ny dörrtyp skapas genom att användaren klickar på knappen ny dörrtyp, skapas ett objekt av typen guideData och läggs in i listan över dörrtyper. Objektet guideData består av en heltalsarray med storlek åtta, en sträng för namnet på dörrtypen samt ett antal metoder. De åtta heltalen i Arrayen representerar var och en av dörrtypens egenskaper samt antalet dörrmiljöer som skall skapas:

Plats

nr	Reperesentation
0	Typ av utsidesläsare
1	Färg på utsidesläsare (om någon)
2	Typ av insidesläsare (om någon)
3	Färg på insidesläsare (om någon)
4	Typ av knapp (om någon)
5	Motorlås (Ja/Nej)

- 6 RC64 (Ja/Nej)
- 7 Antal dörrmiljöer

När dörrtyperna skapas i guiden går det inte att skapa lika många olika typer som det går att skapa direkt i trädet. I guiden kan de dörrmiljöer skapas som i någon mån kan sägas tillhöra de vanligare. En dörrmiljö bestående av en dörrkontrollenhet och två öppnarknappar men inga läsare kan exempelvis inte skapas i guiden, däremot kan den skapas direkt i trädet på flik 1.

4.2.9 Placera ut dörrtyper i trädet

När alla dörrtyper som användaren önskar använda sig av har skapats samt hur många dörrmiljöer som skall skapas av varje dörrtyp angetts, består nästa steg att ta ställning till hur stor del av centralenheternas kapacitet som ska utnyttjas. Poängen med att inte använda centralenheternas fulla kapacitet är att det möjliggör en enklare utbyggnad av passersystemet i ett senare skede där reservkapaciteten hos centralenheterna utnyttjas istället för att nya centralenheter behöver införskaffas. I programmet finns valet att utnyttja 75% av centralenhetens kapacitet, kryssas denna ruta i kommer det sista loopkortet på centralenheten inte att skapas utan istället användas som reservkapacitet.

När användaren klickar på nästa kommer dörrmiljöerna att placeras ut och programmet återgår till flik ett. Om användaren klickar på avbryt kommer dörrmiljöerna inte att placeras ut, men programmet kommer att återgå till flik ett.

Principiellt händelseförlopp när dörrguidens dörrmiljöer placeras ut:

1. De objekt som finns skapade under Flik 1 tas bort.
2. Alla objekt som finns i listan över dörrtyper adderas ihop till en matris. Varje rad i matrisen består av en dörrtyp.
3. De dörrtyper som består av porttelefoner identifieras och placeras ut på centralenheter 9017. Antingen på tre eller fyra loopkort per centralenhet beroende på om 75 % av centralenheternas kapacitet ska användas eller ej.
4. Övriga dörrmiljöer placeras ut, i den ordning de befinner sig i matrisen över dörrtyper.

Tanken med dörrguiden är att skapa ett effektivare sätt att placera ut vanligt återkommande dörrmiljöer. Om användaren utöver dessa miljöer skulle vilja skapa ytterligare dörrmiljöer går det naturligtvis att göra detta i trädet under flik ett. Det går också att byta plats på dörrmiljöerna sedan de lagts till i trädet.

4.2.10 Flik 2

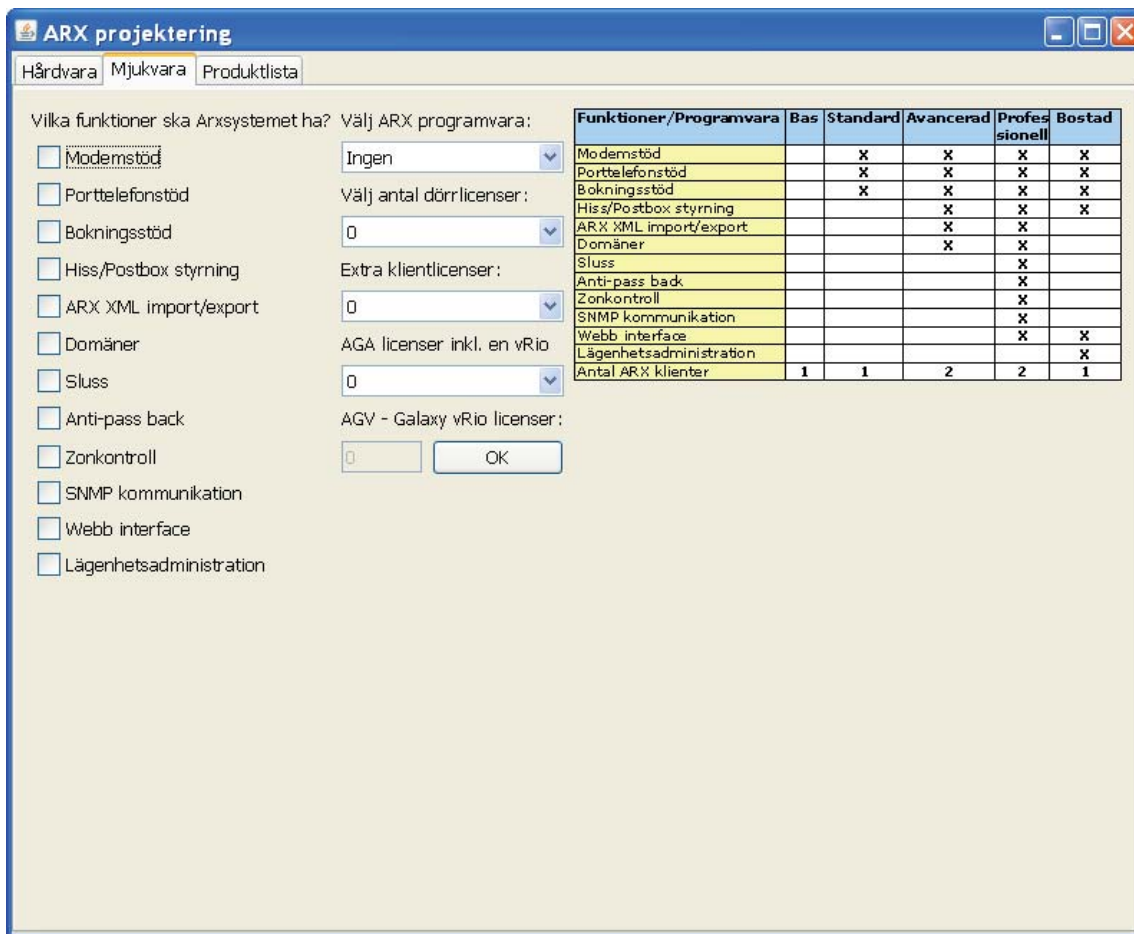


Bild 5, Flik 2.

4.2.11 Programpaket

Flik 2, mjukvara, används för att specificera den mjukvara som ska styra passersystemet. Dels behövs en mjukvara som motsvarar den funktionalitet som önskas i passersystemet. Dels behövs licenser för varje dörrmiljö. För integrering mot Galaxy larmsystem behövs larmintegrationslicenser och ev. så behövs även fler klientlicenser.

Programvaran till ARX passersystem finns i fem olika utföranden, vilket som bör användas beror på vilken funktionalitet i systemet som önskas. Till höger finns en översiktsbild som beskriver vilka funktioner som finns i varje paket. I fältet välj ARX programvara går det att specificera vilket programpaket som önskas, och för att göra valet enklare kan man till vänster specificera vilka funktioner i ARX som önskas och utifrån dessa val välja programvara.

Genom att kryssa i de funktioner som ARX passersystem ska kunna hantera, kan man få en vägledning till vilket programvarupaket man bör välja. Efter att valda funktioner kryssats i blir nämligen programpaketet som inte fungerar med de val som gjorts, röda. Kryssrutorna Porttelefonstöd, Bokningsstöd och Hiss/Postboxstöd är kopplade till hårdvaruträdet på så sätt att om respektive hårdvara läggs till, så blir automatiskt tillhörande kryssruta ikryssad. Om t.ex. en porttelefon läggs till i hårdvaruträdet så kommer kryssrutan Porttelefonstöd automatiskt att bli ikryssad.

Bakom kryssrutorna finns en Array som är lika lång som antalet kryssrutor. När en kryssruta kryssas i, blir motsvarande plats i Arrayen satt till "true". När en kryssruta kryssas ur blir motsvarande plats i Arrayen satt till "false". Efter att en kryssruta kryssats i eller ur, körs en metod som kontrollerar vilka val av programvarupaket som nu är kompatibla/inte kompatibla med funktioner som önskas. När listan över programpaket sedan renderas, kommer de alternativ som inte passar med de valda funktionerna att vara röda.

När ett utav programpaketen valts, skickas information om det valda paketet, vidare till kalkylbladet på flik 3.

4.2.12 Dörrlicenspaket

För varje dörrmiljö (dörrkontrollenhet, kompaktläsare, dörrbladsläsare eller porttelefon) krävs en dörrlicens. Dörrlicenserna säljs i olika stora paket och finns att välja i listan under texten "Välj antal dörrlicenser". Beroende av antal dörrmiljöer i hårdvaruträdet, kommer de licenspaket som innehåller färre antal dörrlicenser än antalet dörrmiljöer i hårdvaruträdet att vara röda. Om t.ex. fem dörrmiljöer är utplacerade i hårdvaruträdet kommer licenspaketet om en och fyra dörrar vara röda.

Varje gång en dörrmiljö läggs till i trädet under flik 1 samt varje gång en dörrmiljö tas bort från trädet kommer programmet att räkna hur många dörrmiljöer som finns kvar. Denna beräkning ligger sedan till grund för hur många licenser som behövs förutsatt att alla dörrmiljöer skall tas i drift.

4.2.13 Röda alternativ

När funktionerna som önskas till programpaketet valts, skulle ett alternativt tillvägagångssätt vara att programmet väljer den programvara som bäst överensstämmer med funktionsvalen som gjorts. Anledningen till att jag inte använt detta alternativ är två. För det första finns det en pedagogisk poäng i att användaren måsta ta ställning till vilket programpaket som ska väljas, dels för att uppmärksamma användaren på att det faktiskt finns flera programpaket dels för att användaren eventuellt lär sig skillnaden mellan programpaketen. Den andra anledningen är att det inte går att utesluta att användaren vill ha ett annat programpaket än det som rekommenderas av programmet. Ett sådant tänkbart scenario skulle kunna vara att användaren väljer att köpa ett programpaket med fler funktioner än vad som angetts för att användaren helt enkelt vill kunna bygga ut systemet i framtiden. När de val som inte rekommenderas av programmet dessutom skrivs ut i rött, bör det ändå stå klart vilka programpaket som inte rekommenderas. Utöver att programmet rekommenderar vissa programpaket, går det även att välja att inte ta något programpaket över huvudtaget. Detta alternativ finns för att det kan tänkas finnas fall där användaren redan äger en licens till en ARX programvara och endast använder detta program som stöd för att bygga ut sitt nuvarande system.

Dörrlicenspaketet fungerar enligt samma logik, inget dörrlicenspaket väljs ut av programmet. Det enda som programmet gör är att avråda användaren från att välja dörrlicenspaket som innehåller färre antal dörrlicenser än utplacerade dörrmiljöer. Genom att tvinga användaren att själv göra ett medvetet val, tydliggörs det faktum att det faktiskt krävs dörrlicenser för de dörrmiljöer som skall tas i drift, samtidigt är det möjligt att välja ett dörrlicenspaket som innehåller färre antal dörrlicenser än

utplacerade dörrmiljöer, eftersom användaren eventuellt inte tänker ta alla dörrmiljöer i drift eller redan äger dörrlicenser. Med andra ord rekommenderas användaren att inte välja paket med för få licenser (röd text), men förbjuds för den skull inte.

4.2.14 Klientlicenser

För varje dator som det ska gå att komma åt ARX ifrån krävs en klientlicens. Beroende på vilket programvarupaket som valts ingår en eller två klient licenser. I paketen Bas, Standard och Bostad ingår en licens och i paketen Avancerad och Professionell ingår två klientlicenser. För att utöver dessa välja fler licenser, anges hur många i fältet ”Extra klientlicenser”.

När ett val gjorts i listan över klientlicenser, skickas information om hur många licenser som valts, vidare till kalkylbladet på flik 3.

4.2.15 Larmlicenser

För att kunna integrera Galaxy larmsystem med ARX passersystem krävs licenser för varje AGA - Galaxy alarmpanel. Till varje AGA ingår en AGV - Galaxy vRio och utöver dessa krävs en licens för vare AGV som skall integreras mot ARX passersystem. Ange antal AGA licenser och utöver de AGV licenser som följer med AGA, ange antal extra AGV licenser.

När ett val gjorts i listan över AGA skickas information om hur många licenser som valts, vidare till kalkylbladet på flik 3. Om antalet AGA licenser som valts, inte är noll, kommer fältet där AGV licenser kan anges, att tändas. När ett antal matats in kommer, informationen om hur många AGV licenser som inmatats att skickas vidare till kalkylbladet på flik 3. Om det som matas in i AGV fältet inte är ett positivt heltal kommer programmet att tolka inmatningen som noll. Anledningen till att AGA väljs ur en lista emedan AGV får anges fritt är att det lär användas betydligt fler AGV än AGA.

4.2.16 Flik 3

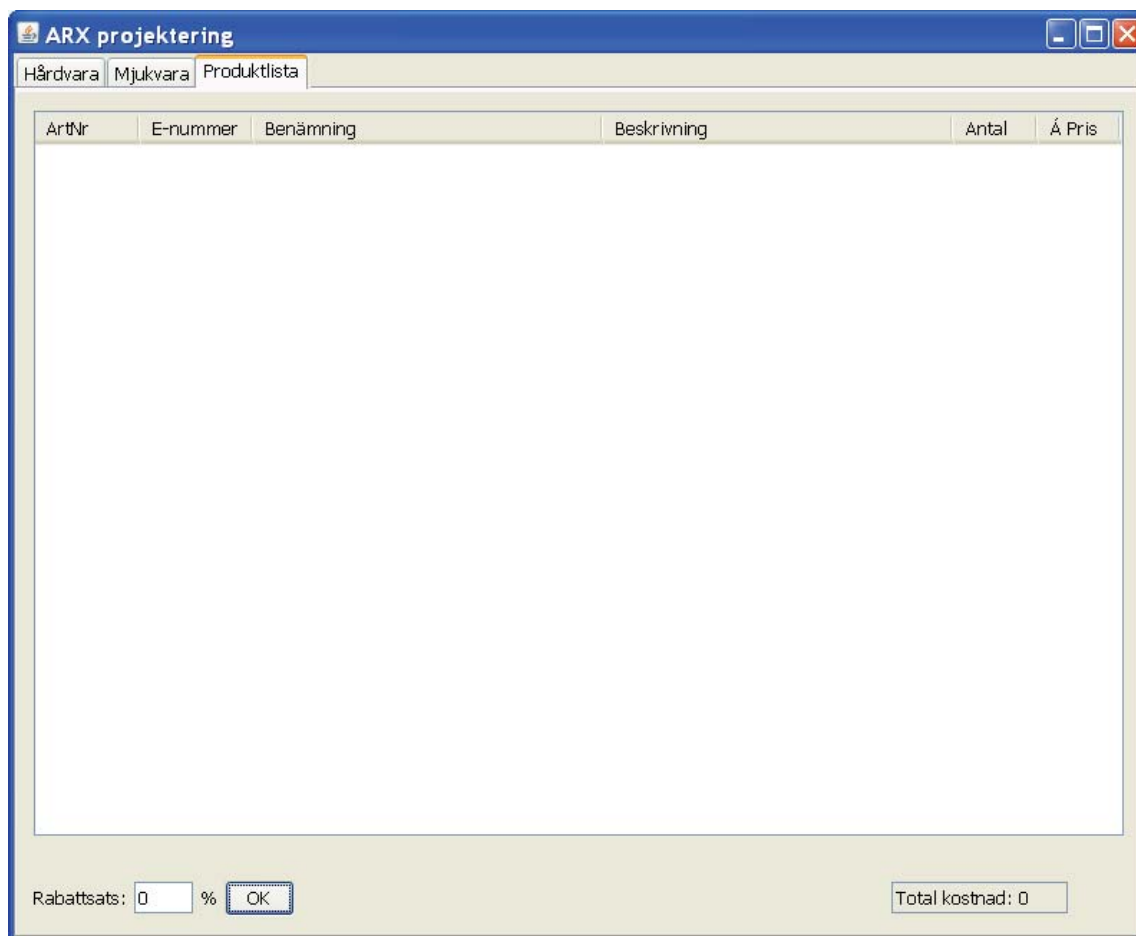


Bild 6, Flik 3.

Flik 3 består av ett kalkylblad, ett fält för att mata in rabattsats, en knapp och ett fält där den totala kostnaden för passersystemet visas. Kalkylbladet är en sammanställning av produkterna som lagts till under flik 1 och mjukvaran som lagts till under flik 2. Här visas produkterna med artikel nummer, E-nummer, benämning, beskrivning, antal och pris.

4.2.17 Datatypen tableData

För alla produkter som skapas eller tas bort under flik 1 samt alla mjukvaruprodukter som specificeras under flik 2 skickas information till kalkylbladet i flik 3. Kalkylbladet skapar sedan, för varje typ av produkt, ett objekt av typen tableData. Varje objekt av typen tableData innehåller information om produktens Artikelnummer, E-nummer, namn, beskrivning, antal, pris och typ. Alla objekt av typen tableData läggs efter att de skapats in i en ArrayLista, tableList. Skillnaden mot ArrayListan Hardware är att endast ett objekt av varje typ kommer att skapas, exempelvis kommer två objekt av typen Centralenhet 9016 inte att finnas. Istället återspeglas det att det finns två av denna produkt genom egenskapen Antal.

4.2.18 Lägga till och ta bort en komponent

Hur en produkt läggs till beror på om det är en komponent från Flik 1 eller från Flik 2. Produkterna som specificeras under flik två kan endast existera i ett exemplar.

Exempelvis innebär att ett 16 dörrars licenspaket specificeras, att inget annat dörrpaket är specificerat och om ett dörrlicenspaket var specificerat tidigare så ska det tas bort. När det gäller produkter från flik ett kan flera likadana produkter finnas samtidigt.

Principiellt händelseförlopp för att lägga till en produkt från Flik 1 i kalkylbladet:

1. Metoden för att lägga till en produkt av typ Flik 1 anropas med produkten och dess pris.
2. tableList söks igenom för att kolla om en produkt av samma typ redan finns.
3. Om en produkt av samma typ hittas ökas egenskapen Antal med ett.
4. Om produkten inte finns i tableList skapas objektet och läggs in i tableList.

Principiellt händelseförlopp för att ta bort en produkt från flik ett i kalkylbladet:

1. Metoden för att ta bort en produkt av typ Flik 1 anropas med produktens typnummer.
2. tableList söks igenom för att hitta produktens plats.
3. Om egenskapen Antal är större än ett, minskas antalet med ett
4. Om egenskapen Antal är precis ett, tas objektet bort.

Principiellt händelseförlopp för att lägga till och/eller ta bort en produkt från Flik 2 i kalkylbladet:

1. Metoden för att lägga till/ta bort en produkt av typ Flik 2 anropas med produktens artikelnummer, E-nummer, namn, beskrivning, antal, pris och typ.
2. tableList söks igenom, och hittas en produkt av samma typ, tas denna bort.
3. Produkten skapas och läggs in i tableList.

4.2.19 Systemets kostnad

Fältet till höger under kalkylbladet visar passersystemets totala kostnad, baserat på vad som finns i kalkylbladet. Eftersom återförsäljare av ARX passersystem ofta har en fast rabattsats på alla produkter från ASSA finns en möjlighet att till vänster ange en rabattsats för att på så sätt räkna ut en annan totalkostnad. Rabattsatsen måste anges som ett heltal mellan 0 och 100.

4.2.20 Använda priser

Eftersom priser på produkter kan komma att ändras, finns en möjlighet att i detta program importera en ny uppsättning priser. Importen av priser görs alltid när programmet startar. Om prisimporten misslyckas kommer 2008 års priser att användas.

Principiellt händelseförlopp när priser importeras

1. När programmet startas kollar programmet om filen Prislsta.skv ligger i samma katalog som programmet.
2. Om filen Prislsta.skv inte hittas används 2008 års priser. Om filen hittas kommer den att öppnas och importen startas.
3. På alla rader tas följande bort: mellanslag, S och kr.
4. För varje rad kommer de första tecknen att läsas in,
 - a. om dessa tecken bildar ett heltal som matcher artikelnumret hos någon utav de produkter som finns inlagda i systemet,
 - b. och detta tal är följt av ett semikolon
 - c. och sedan ett nytt heltal,

- d. följt av ett semikolon
- 5. Så kommer det senare heltalet att användas som priset på produkten med det matchande artikelnumret.

4.2.21 Exempel

Om en rad i Prislista.skv börjar med "S554 014 184;2 780 kr;" kommer programmet att göra om raden så att det står "554014184;2780;". Om programmet hittar en produkt med artikel nummer 554014184 så kommer produkten att tilldelas priset 2780. Anledningen till att just mellanslag, S och kr tas bort beror på att prislistan i dagsläget, när den sparas som en semikolonseparerad fil ser ut precis som i exemplet ovan.

5 Slutsatser

Sammanfattningsvis, med utgångs punkt att skapa ett verktyg som förenklar förståelsen för- och projekteringsarbetet av- ARX passersystem, har jag med hjälp av Java och det Vicente kallar formativitet designat detta program. Även om jag med detta program kommit en bit på vägen, finns mycket kvar att göra på detta område, exempelvis att flytta fler parametrar utanför programmet. Jag hade i början som utgångspunkt att alla typer av hårdvaruprodukter skulle specificeras utanför programmet, exempelvis i en excel-fil. Detta visade sig dock mer komplicerat än jag först räknat med, framför allt eftersom vissa utav produkterna har en ganska komplicerad logik för vad som går att addera och hur.

5.1 Kalkylering och Projektering

Under projektets gång har två olika typer av potentiella användare mer och mer utkristalliserats. Dels den typ av användare som jobbar med att ta reda på ett pris på hur mycket hela anläggningen skulle komma att kosta och dels den typ av användare som har till uppgift att få hela systemet att fungera.

5.1.1 Användartyp 1

Denna typ av användare är i regel intresserad av att få med alla beståndsdelar av passersystemet, även det som inte direkt säljs av passersystemsleverantören såsom lås, kablar och kringutrustning och utifrån detta skapa en offert. En annan viktig del i offerten är tidsåtgången, d.v.s. hur lång tid installationen beräknas ta. För att denna ska kunna beräknas, skulle det behöva finnas en möjlighet att ange hur tid varje enhet beräknas ta att montera och driftsätta. En annan sak som skulle behövas inkluderas är kabeldragningen mellan enheterna. Hur lång tid den beräknas ta beror på två saker, dels avståndet mellan enheterna och dels hur svårt det är att dra kabeln. Att dra en kabel på en kabelstege ovanför ett innertak går exempelvis snabbare än om den måste dras genom en vägg som först måste borraras upp.

5.1.2 Användartyp 2

Den typ av användare som har till uppgift att få hela systemet att fungera har inte samma informationsbehov som användaren ovan. Det centrala för denna typ av användare är istället strömförsörjningen. Även om passersystem i sig inte har några säkerhetskrav på driftsäkerhet så får dom indirekt det om dom integreras med larmsystem. Ett krav på larmsystem som ska uppnå en viss typ av larmklassning är exempelvis hur länge systemet skall kunna vara i drift en viss tid vid strömavbrott. Med andra ord finns här ett behov av att veta hur stor strömförbrukningen är så att batterierna som skall fungera som backup kan dimensioneras rätt. En enhet har i regel olika strömförbrukning när dom används och när dom inte används. Ett motorlås drar mer ström när regeln ändrar läge än när den inte gör det. Sett till dörren som helhet är antalet passager genom dörren helt avgörande för hur stor strömförbrukningen är. Ytterligare en sak som måste tas hänsyn till här är avståndet mellan enheterna och vilka dimensioner kabeln som dras mellan enheterna har.

5.1.3 Användartyp 3

En tredje typ av användare skulle kunna klassificeras som passersystemsåterförsäljare. Denna typ av användare, som delvis liknar användartyp 1, intresserar sig för att få med alla delar av passersystemet och vad dessa kostar, men eftersom denna typ av användare endast säljer systemet vidare till den som i sin tur ska installera systemet behöver denne inte ta med tidsåtgångsaspekten. Denna användartyp har istället ett behov av att så enkelt och snabbt som möjligt få fram en specificerad lista på vad passersystemet ska bestå av för komponenter.

5.1.4 Inriktning på en grupp

Så som projekteringsverktyget är utformat i dagsläget ligger det närmast användartyp 3. Jag tror att vid en uppföljning/vidareutveckling av användarhjälpmedlet så skulle även en eller båda de andra användartyperna kunna integreras. Samtidigt finns det en risk att när verktyget växer i komplexitet så minskar samtidigt graden av användbarhet som verktyget har.

6 Referenser

Bergquist, Johan. 1996. *Vem kan man lita på?. Om minnets exakthet och motståndskraft mot påverkan* <http://oldwww.cs.umu.se/tdb/kurser/TDBC12/HT-96/LABBAR/johan.bergquist/index.html> (17-02-08).

Gulliksen, Jan & Göransson Bengt. 2002. *Användarcentrerad systemdesign*. Studentlitteratur: Lund.

Java. . <http://www.sun.com/java/about/index.jsp> (17-02-08),
<http://java.sun.com/javase/6/docs/api/javax/swing/GroupLayout.html> (17-02-08),
<http://java.sun.com/docs/books/tutorial/uiswing/components/tree.html> (17-02-08).

Molina, Irene. 1997. *Stadens rasifiering. Etnisk bostadssegregation i folkhemmet*. Geografiska regionstudier 32. Uppsala: Institutionen för kulturgeografi.

Nygren, Else. 1997. Grafiska användargränssnitt, några tips.
<http://www.it.uu.se/research/hci/publications/papers/60/60.html> (17-02-08).

Vicente, Kim. 1999. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Lawrence Erlbaum Associates, Publishers: USA.

ARX projekteringsverktyg - inledning

Om ARX projekteringsverktyg

Denna manual beskriver funktionerna som finns i applikationen ARX projekteringsverktyg. Applikationen består av tre flikar och tanken är att man ska gå igenom programmet från flik 1 via flik 2 och slutligen till flik 3 men det går även att gå tillbaka och ändra samt ta bort och lägga till komponenter i efterhand.

Applikationens syfte tvådelat. Dels syftar applikationen till att underlätta förståelsen av hur ett ARX passersystem byggs upp, vilka komponenter som kan ingå samt hur kopplingen mellan hårdvara och mjukvara ser ut. Dels syftar applikationen till att sammanställa en produktlista över alla komponenter som specificerats till passersystemet.

Priserna som används i applikationen tas ur filen Prislista.skv som ligger i samma katalog som applikationen själv. Om filen Prislista.skv inte ligger i applikationens katalog används 2008 års priser.

Flik 1, Hårdvara

Om flik 1

Flik 1, hårdvara används för att bygga upp hårdvarukomponenterna till ett ARX passersystem, det är denna flik som först visas när programmet startas. När programmet startas finns endast en ikon i det övre högra hörnet, Site. När fler komponenter läggs till kommer dom att placeras i ett träd så att deras inbördes hierarkiska ordning blir tydlig.

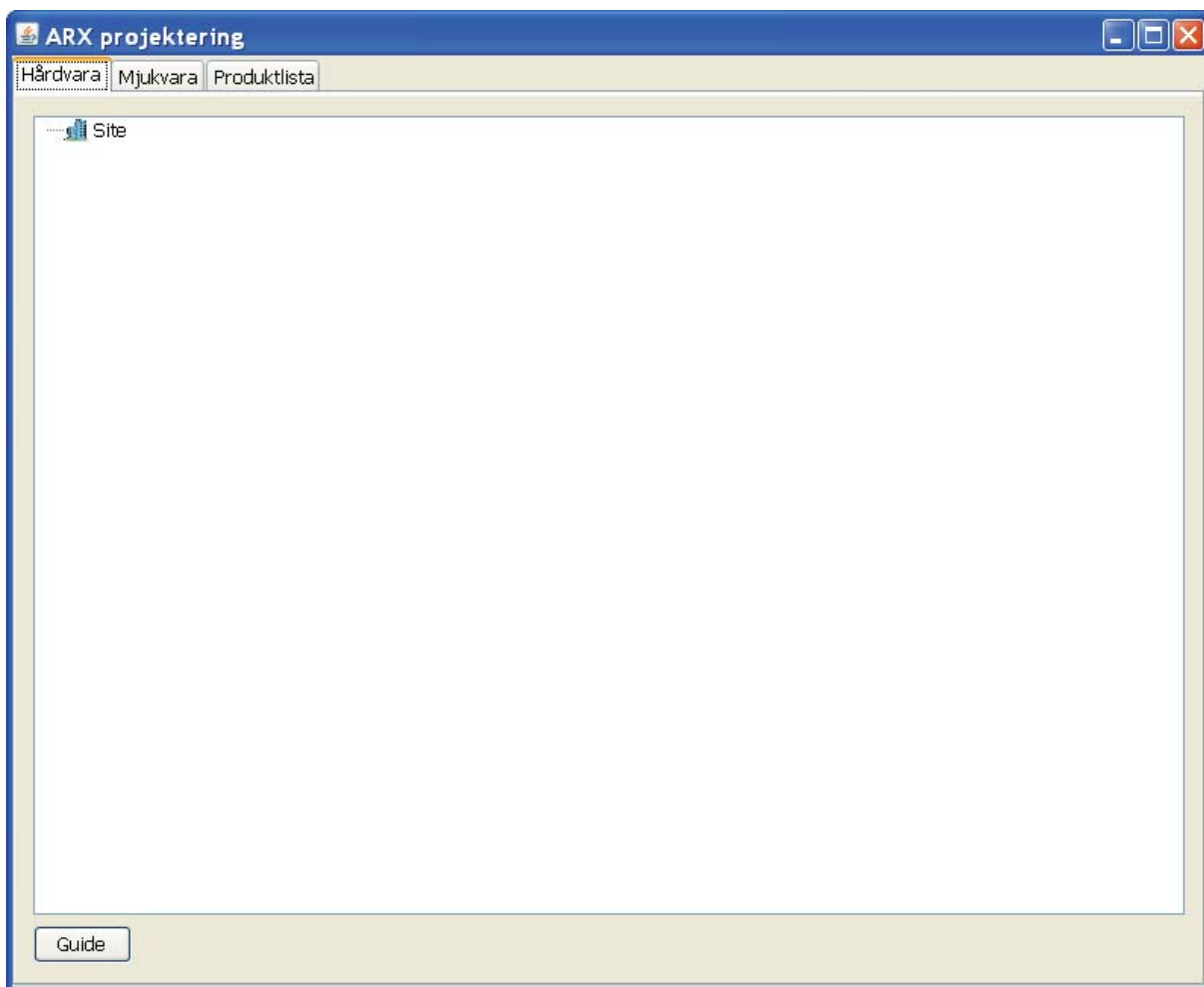


Bild 1, Flik 1, Hårdvara

Lägga till en komponent

Alla komponenter som läggs till kommer att vara kopplade till en redan befintlig enhet. Det är med andra ord inte möjligt att lägga till en läsare utan att det hör till någon av de befintliga

dörrkontrollenheterna. Undantaget är komponenten ”Site” som finns inlagd från början och heller inte går att ta bort.

För att lägga till en komponent:

1. Högerklicka på en befintlig komponent
2. Välj en utav de komponenter som presenteras i popup-menyn
3. Om den befintliga enheten inte redan är full kommer den valda enheten att skapas och höra till den enhet som högerklickades.

Flytta en komponent

Att flytta en komponent från en plats till en annan är endast möjligt om, dom normalt sett kan läggas till varandra och om det finns plats.

För att flytta en komponent från en plats till en annan:

1. Markera den enhet som ska flyttas
2. Dra komponenten till önskat ställe, om enheten kan flyttas dit kommer musperkaren att se ut som en pil med en underliggande fyrkant. Om enheten inte går att flytta dit musperkaren är, kommer pekaren att anta formen av en cirkel med ett streck. Vid en misslyckad flyttning kommer enheten att ligga kvar där den först drogs från.

Ta bort en komponent

Alla komponenter utom komponenten ”Site” går att ta bort. Om en komponent, som har andra komponenter kopplade till sig, tas bort, kommer även de underliggande komponenterna att tas bort.

För att ta bort en komponent:

1. Högerklicka på komponenten som ska tas bort
2. Välj alternativet längst ner, Ta bort...
3. Komponentens tas nu bort.

Komponent-kompatibilitetsschema

Följande tabell visar vilka och hur många komponenter som går att koppla ihop samt hur många.

Exempelvis kan en dörrkontrollenhet DAC420 kopplas ihop med totalt fyra läsare, och oberoende av antal läsare dessutom kopplas ihop med totalt tre öppnarknappar.

Produkt	Inkopplingsbara komponenter	Antal
Site	Centralenhet 9016	
	Centralenhet 9017	Obegränsat
Centralenhet 9016	Loopkort 4014	Tre stycken
	Reläkort 6416	Ett
Centralenhet 9017	Bokningspanel 8001PL	En, men inte om RCE enhet finns på undercentralen
	Loopkort 9018	
	Loopkort 4014	Totalt tre stycken
	Reläkort 6416	Ett
Loopkort 9014	Bokningspanel 8001PL	En, men inte om RCE enhet finns på undercentralen
	DAC 420	
Loopkort 4014	DAC 430	
	PCR 45 (Vit/Svart)	

	PCR 40 (Vit/Svart) PCR 34 (Vit/Svart) DBL 34 (Höger/Vänster ut)	Totalt fyra stycken
	RCE-16	Fyra stycken, men inte om Bokningspanel 8001PL finns på undercentralen
Loopkort 9019	ECP 30	Två stycken
Loopkort 9018	RCE-16	Två stycken, men inte om Bokningspanel 8001PL finns på undercentralen
DAC 420	Läsare 6354 (Vit/Svart)	
DAC 430	Läsare 6355 (Vit/Svart)	
	Läsare 6480	
	Läsare 6481	
	Läsare 6485 (Vit/Svart)	
	Läsare 6585 (Vit/Svart)	
	Interface 6390IF	Totalt fyra stycken
	Öppnarknapp TKN-01	
	Öppnarknapp TKN-02	
	Öppnarknapp TKN-40	
	Öppnarknapp TKN-60	Totalt tre stycken
	Reläkort RC64	
	Reläkort RW22	Ett kort
PCR 45 (Vit/Svart)	Öppnarknapp TKN-01	
PCR 40 (Vit/Svart)	Öppnarknapp TKN-02	
PCR 34 (Vit/Svart)	Öppnarknapp TKN-40	
ECP 30	Öppnarknapp TKN-60	Totalt tre stycken

Tabell 1, komponentschema

Dörrguide

Om dörrguide

Ett alternativt sätt att lägga till komponenter är genom att använda dörrguiden. När dörrguiden används definierar användaren ett antal dörrtyper som ska ingå i passersystemet. När dörrtyperna är definierade samt hur många av varje dörrtyp som ska ingå i passersystemet kommer programmet att placera ut alla dörrmiljöer på centralenheter och loopkort i hårdvaruträdet. Användaren kan sedan justera hur dörrmiljöerna ska fördelas över centralenheterna och/eller skapa ytterligare dörrmiljöer direkt i hårdvaruträdet. Att använda dörrguiden kan vara ett effektivt sätt att placera ut komponenter om flera identiska dörrmiljöer ska placeras i hårdvaruträdet. Observera dock att komponenter som lagt till innan dörrguiden används kommer att tas bort när dörrmiljöerna skapade i dörrguiden placeras ut.

För att komma till dörrguiden, klicka på knappen "Guide" längst ner på flik 1.

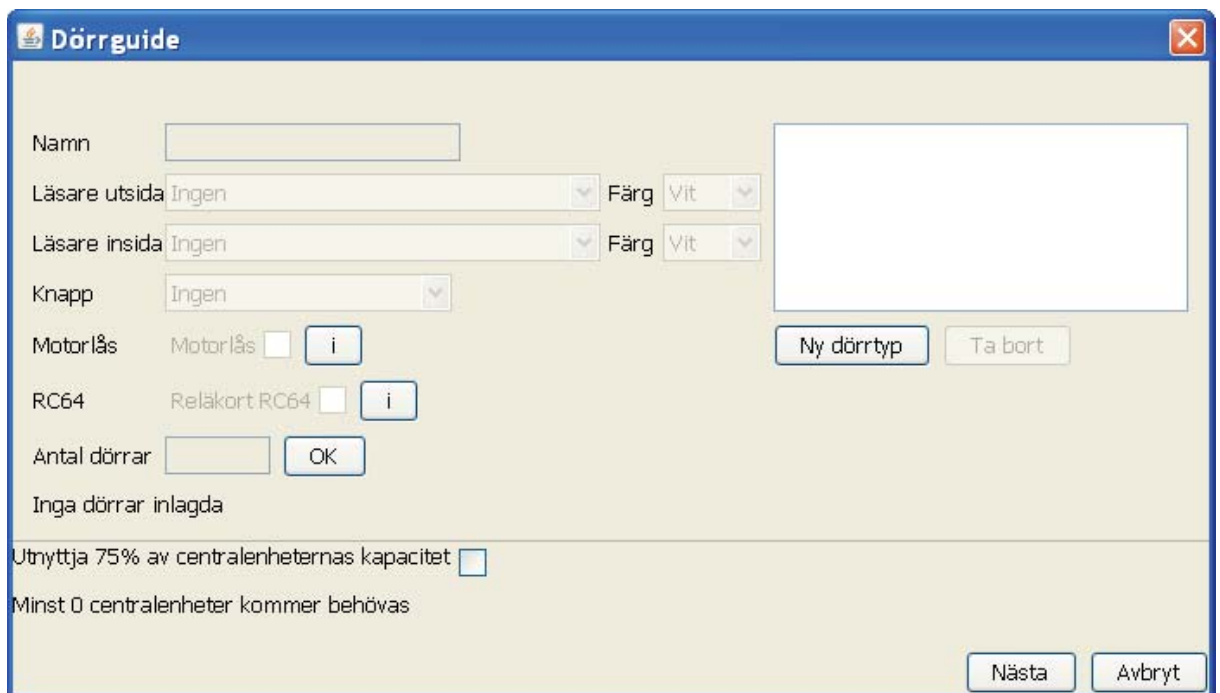


Bild 2, Dörrguiden utan skapade dörrmiljöer

Skapa en dörrtyp

Det finns ingen egentlig begränsning i applikationen för hur många dörrtyper som kan skapas.

För att skapa en ny dörrtyp:

1. Klicka på knappen "Ny dörrtyp". En ny dörrtyp kommer nu att skapas i listan över dörrtyper och fälten "Namn", "Läsare utsida" och "Antal dörrar" kommer att tändas.
2. Skriv in ett namn på dörrtypen i fältet "Namn".
3. Ange en vilken typ av läsare som skall sitta på dörrtypens utsida genom att välja ett utav alternativen i fältet "Läsare utsida". Valet av utsideläsare kommer sedan att styra dom övriga fälten. Emedan valet av utsideläsare är obligatoriskt är alla efterföljande val valfria.
 - Om en Mifare, beröringsfri eller magnetkortsläsare valts som utsideläsare kommer alla andra fält tändas, och när dessa dörrmiljöer placeras ut kommer dom att kopplas till en dörrkontrollenhet. Om läsaren finns i fler än en färg kommer även detta fält att tändas.
 - Om en utav kompaktläsarna eller porttelefon ECP30 valts kommer endast fältet "Knapp" att tändas. Då alla kompaktläsare finns i två färger kommer även fältet "Färg" att tändas.
 - Om dörrbladläsaren DBL34 Höger ut/Vänster ut valts kommer inga fler fält att tändas då inga andra enheter kan kopplas till denna.
4. Om fältet "Läsare insida" är tänt, välj, om så önskas, en insideläsare.
5. Om fältet "Knapp" är tänt, välj, om så önskas, en öppnarknapp för dörrtypen.
6. Kryssrutan "Motorlås" är tänd om en dörrkontrollenhet kommer att användas för dörrtypen. Kryssas denna ruta i kommer dörrkontrollenhet DAC430 att användas. Lämnas denna ruta oikryssad kommer dörrkontrollenhet DAC420 att användas.
7. Kryssrutan "Reläkort RC64" är tänd om en dörrkontrollenhet kommer att användas för dörrtypen. Reläkort RC64 ger dig bland annat tillgång till fler ut- och ingångar på dörrkontrollenheten. Kryssas i denna ruta för att lägga till ett reläkort RC64 på denna dörrtyp.
8. Ange sedan hur många dörrmiljöer du vill skapa av denna dörrtyp och klicka på OK.

Det totala antalet dörrar som skapats kommer att visas under fältet "Antal dörrar"

Ändra en dörrtyp

För att ändra utseendet på en dörrtyp, eller antalet dörrmiljöer som skall skapas av dörrtypen:

1. Markera dörrtypen i listan över dörrtyper
2. Förändra dörrtypen och/eller ange antal dörrmiljöer som skall skapas
3. Klicka på OK knappen till höger om fältet antal dörrar

Ta bort en dörrtyp

För att ta bort en dörrtyp:

1. Markera dörrtypen i listan över dörrtyper
2. Klicka på Ta bort knappen

Placera ut dörrmiljöerna

Innan dörrmiljöerna placeras ut på hårdvaruträdet ange om 75% av centralernas kapacitet ska utnyttjas eller ej. När denna ruta är ikryssad kommer dörrmiljöer endast placeras ut på 3 av centralenhetens loopkort, detta för att i ett senare skede möjliggöra en utbyggnad av systemet.

För att placera ut dörrmiljöerna och återgå till hårdvaruträdet, klicka på nästa. För att inte placera ut dörrmiljöerna men återgå till hårdvaruträdet, klicka på avbryt.

Flik 2, Mjukvara

Om flik 2

Flik 2, Mjukvara används för att välja mjukvarulicenser som motsvarar, den funktionalitet som användaren önskar och den hårdvara som lagts till under flik 1. De licenser som går att specificera är programvarupaket, dörrlicenspaket, klientlicenser och Galaxy larmintegrationslicenser.

ARX projektering

Hårdvara Mjukvara Produktlista

Vilka funktioner ska Arxsystemet ha? Välj ARX programvara:

- Modemstöd
- Porttelefonstöd
- Bokningsstöd
- Hiss/Postbox styrning
- ARX XML import/export
- Domäner
- Sluss
- Anti-pass back
- Zonkontroll
- SNMP kommunikation
- Webb interface
- Lägenhetsadministration

Välj antal dörrlicenser: 0

Extra klientlicenser: 0

AGA licenser inkl. en vRio: 0

AGV - Galaxy vRio licenser: 0

OK

Funktioner/Programvara	Bas	Standard	Avancerad	Professionell	Bostad
Modemstöd		X	X	X	X
Porttelefonstöd		X	X	X	X
Bokningsstöd		X	X	X	X
Hiss/Postbox styrning			X	X	X
ARX XML import/export			X	X	
Domäner			X	X	
Sluss				X	
Anti-pass back				X	
Zonkontroll				X	
SNMP kommunikation				X	
Webb interface				X	X
Lägenhetsadministration					X
Antal ARX klienter	1	1	2	2	1

Bild 3, Flik 2, Mjukvara

ARX programvara

Programvaran till ARX passersystem finns i fem olika utföranden, vilket som bör användas beror på vilken funktionalitet i systemet som önskas. Till höger finns en översiktstabell som beskriver vilka funktioner som finns i varje paket. I fältet välj ARX programvara går det att specificera vilket

programpaket som önskas, och för att göra valet enklare kan man till höger specificera vilka funktioner i ARX som önskas.

För att välja ett programvarupaket:

1. Kryssa i den eller de funktioner som du vill att programmet skall kunna hantera. Kryssrutorna Porttelefonstöd, Bokningsstöd och Hiss/Postboxstöd är kopplade till hårdvaruträdet på så sätt att om respektive hårdvara läggs till, så blir automatiskt tillhörande krysstura ikryssad. Om t.ex. en porttelefon läggs till i hårdvaruträdet så kommer kryssrutan Porttelefonstöd automatiskt att bli ikryssad.
2. Välj ett av programvarualternativen i fältet "Välj ARX programvara". Baserat på de val som gjorts i steg 1 kommer sedan vissa av i listan till fältet "Välj ARX programvara" att bli röda. De alternativ som är röda uppfyller inte de funktioner som specificerats med kryssrutorna till vänster.

Dörrlicenser

För varje dörrmiljö (dörrkontrollenhet, kompaktläsare, dörrbladsläsare eller porttelefon) krävs en dörrlicens. Dörrlicenserna säljs i olika stora paket och finns att välja i listan till fältet "Välj antal dörrlicenser". Beroende av antal dörrmiljöer i hårdvaruträdet, kommer de licenspaket som innehåller färre antal dörrlicenser än antalet dörrmiljöer i hårdvaruträdet att vara röda. Om t.ex. fem dörrmiljöer är utplacerade i hårdvaruträdet kommer licenspaketen om en och fyra dörrar vara röda.

Klientlicenser

För varje dator som ska gå att komma åt ARX ifrån krävs en klientlicens. Beroende på vilket programvarupaket som valts ingår en eller två klient licenser. I paketen Bas, Standard och Bostad ingår en licens och i paketen Avancerad och Professionell ingår två klient licenser. För att utöver det välja fler licenser, ange hur många i fältet "Extra klientlicenser".

Larmlicenser

För att kunna integrera Galaxy larmsystem med ARX passersystem krävs licenser för varje AGA - Galaxy alarmpanel. Till varje AGA ingår en AGV - Galaxy vRio och utöver dessa krävs en licens för vare AGV som skall integreras mot ARX passersystem. Ange antal AGA licenser och utöver de AGV licenser som följer med AGA, ange antal extra AGV licenser och klicka på OK.

Flik 3, Produktlista

Om flik 3

Flik 3 är en sammanställning av produkterna som lagts till under flik 1 och mjukvaran som lagts till under flik 2. Här visas produkterna med artikel nummer, E-nummer, benämning, beskrivning, antal och pris.

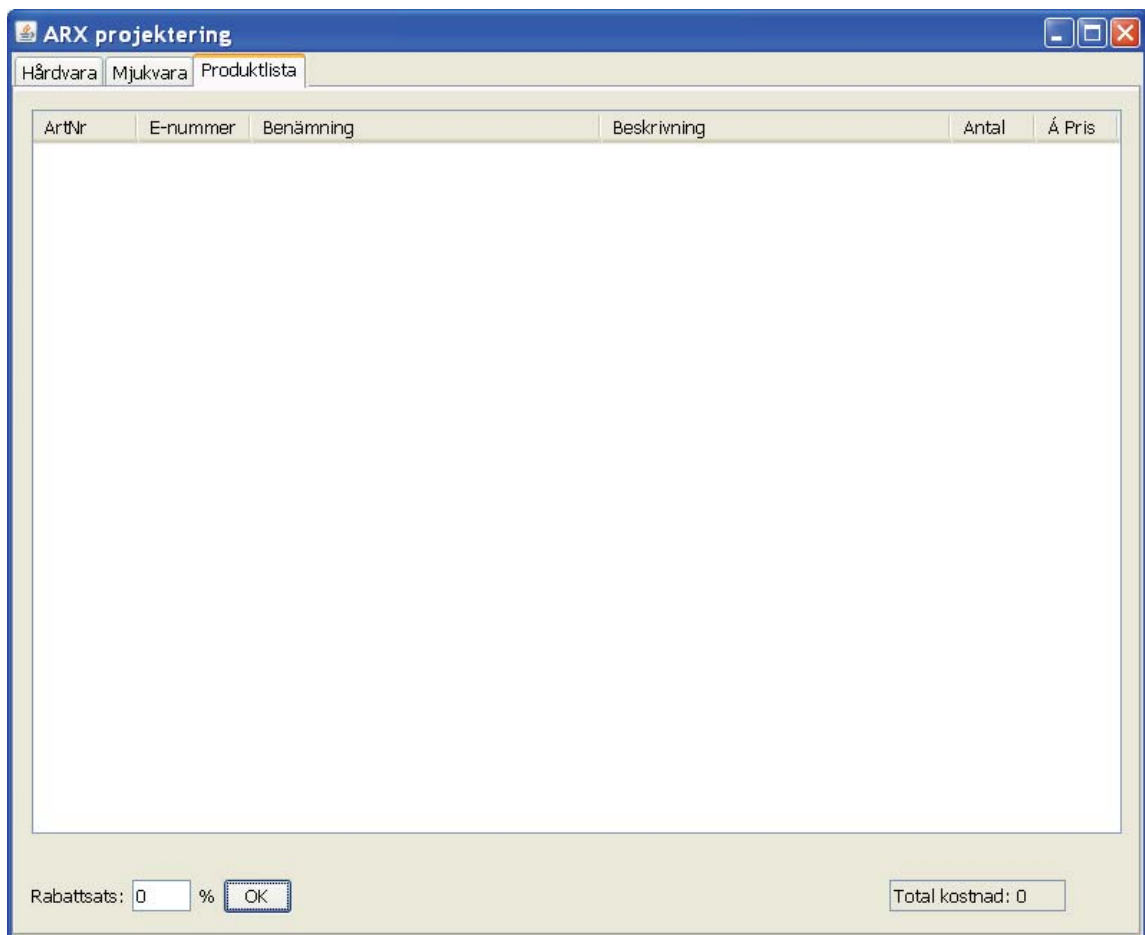


Bild 4, flik 3, Produktlista

Kostnad

För att räkna ut nettokostnaden för det specificerade passersystemet:

1. Ange eventuell rabattsats i hela procent
2. Klicka på OK
3. Nettopriset presenteras nu i det nedre högra hörnet.