

Datavetenskap

---

**Martin Blomgren**

**Roger Andersson**

**Utveckling av grafiska terminaler för  
processindustrin**

---

Examensarbete, C-nivå

2002:30



# **Utveckling av grafiska terminaler för processindustrin**

**Martin Blomgren**

**Roger Andersson**



Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

---

Martin Blomgren

---

Roger Andersson

Godkänd, 2002-06-14

---

Handledare: Thijs Jan Holleboom

---

Examinator: Tim Heyer



## **Sammanfattning**

Denna rapport beskriver utvecklingen av programvara för grafiska övervakningsterminaler till Bravikens pappersbruk. Dessa terminaler ska visa status på maskin enheter och vart pappersrullarna befinner sig i en maskinhall. Förutom detta skulle en del funktionalitet implementeras för styrning av en PLC, *Programmable Logic Controller*. Det är nämligen denna enhet som styr alla maskiner.

Syftet med systemet är att underlätta övervakningen och styrningen av det maskinområde som finns i magasin 3. Då systemet är uppbyggt enligt klient/server principen kan man även övervaka magasin 3 från operatörsstationer som finns utplacerade på andra platser i fabriken.

Utvecklingen av terminalprogrammet har gjorts med hjälp av Siemens WinCC 5.0 för processvisualisering och Siemens Step 7 5.1 för styrning av PLC datorerna.

# Development of graphical terminals for the process industry

## Abstract

This report will describe the development of software for graphical monitoring terminals for Braviken paper mill. These terminals will show status of machine units and locations for each paper roll that's currently in the machine hall. Also, some functionality for controlling a PLC, *Programmable Logic Controller*, should be implemented.

The systems purpose is to facilitate supervision and control of the machine area in warehouse 3. As the system is built upon the client/server principle it is also possible to supervise and control warehouse 3 from other locations in the factory.

The development of the software was made in Siemens WinCC 5.0 for process visualization and Siemens Step 7 5.1 for control of the PLC units.



# Innehållsförteckning

<b>1</b>	<b>Generell struktur</b> .....	<b>1</b>
<b>2</b>	<b>Inledning</b> .....	<b>2</b>
2.1	Information och historik om Bravikens pappersbruk .....	3
2.2	Rullhantering .....	3
2.2.1	Rullhantering fram till magasinering .....	
2.2.2	Rullhantering i magasinet .....	
<b>3</b>	<b>Förutsättningar och utvecklingsverktyg</b> .....	<b>5</b>
3.1	Utvecklingsmiljö.....	5
3.2	Beskrivning av WinCC.....	5
3.2.1	Delprogram .....	
3.2.2	WinCC Basic Options .....	
<b>4</b>	<b>Utveckling av systemet</b> .....	<b>10</b>
4.1	Syfte.....	10
4.2	Omfattning.....	10
4.3	Projektplanering.....	10
4.3.1	Tidsplan .....	
4.3.2	Kravspecifikation .....	
4.4	Beskrivning av konstruktionslösningen.....	12
4.4.1	Tagbeskrivning .....	
4.4.2	Uppsättning av taggar .....	
4.4.3	Terminalöversikt .....	
4.4.4	Statiska bilder .....	
4.4.5	De dynamiska delarna i en bild .....	
4.4.6	Pappersrullar med dynamisk storlek .....	
4.4.7	Pappersrullar med fast storlek .....	
4.4.8	Allmänt för samtliga pappersrullar .....	
4.4.9	Protokoll för visning av rullinformation .....	
4.4.10	Byte av språk .....	
4.4.11	Larmhantering .....	
<b>5</b>	<b>Test av systemet</b> .....	<b>24</b>
<b>6</b>	<b>Problem under konstruktion</b> .....	<b>25</b>
6.1	Installation och inläring .....	25
6.2	Uppsättning av nätförbindelse .....	25

6.3	Övriga problem.....	26
6.3.1	Projektmiljö	
6.3.2	Utvecklingsverktyg	
<b>7</b>	<b>Erfarenheter .....</b>	<b>28</b>
<b>8</b>	<b>Slutsatser.....</b>	<b>29</b>
	<b>Referenser .....</b>	<b>30</b>
<b>Bilagor</b>		
<b>A</b>	<b>Ritningar och bilder.....</b>	<b>31</b>
A.1	Fabriksöversikt .....	31
A.2	Översikt övervåning.....	32
A.3	Översikt undervåning.....	33
A.4	Nätverksöversikt.....	34
<b>B</b>	<b>Enhets nummer .....</b>	<b>35</b>
<b>C</b>	<b>Protokollspecifikationer .....</b>	<b>36</b>
C.1	Brevlåda för rullinformation.....	36
C.1.1	Struktur för brevlådan	
C.1.2	Användning	
<b>D</b>	<b>Kravs specifikation .....</b>	<b>37</b>
D.1	Inledning.....	37
D.1.1	Allmänt	
D.2	Översikt.....	37
D.2.1	Problem som systemet ska lösa	
D.2.2	Omgivande systems miljö	
D.3	Systembeskrivning.....	38
D.3.1	Funktioner	
D.3.2	Prestandakrav	
D.3.3	Övrigt	
<b>E</b>	<b>Lamb AGs riktlinjer för WinCC projekt .....</b>	<b>42</b>
E.1	Uppbyggnad.....	42
E.2	Lager42	
<b>F</b>	<b>Systembilder under körning.....</b>	<b>43</b>
F.1	Översikt, startbild .....	43
F.2	Sorteringsbordet.....	44
F.3	Utmatning .....	45
F.4	Hydraulik.....	46
F.5	Informations visning av pappersrulle .....	46

<b>G Begrepp .....</b>	<b>47</b>
<b>H Användarmanual.....</b>	<b>49</b>

## Figurförteckning

Figur 1: WinCCExplorer overview .....	6
Figur 2: Ex Strukturer .....	13
Figur 3: Ex Instans av struktur .....	14
Figur 4: Terminalöversikt .....	15
Figur 5: Ex Statisk bild .....	16
Figur 6: Ex Statisk bild med dynamiska objekt .....	17
Figur 7: Dynamisk storlek på rullar .....	18
Figur 8: Sprängskiss av mynt .....	18
Figur 9: Rullinformation .....	20
Figur 10: Alarm Logging editorn .....	23
Figur 11: Översikt Braviken .....	31
Figur 12: Översikt övervåning .....	32
Figur 13: Översikt undervåning .....	33
Figur 14: Nätverksöversikt .....	34
Figur 15: Översikt, startbild under körning .....	43
Figur 16: Sorteringsbord under körning .....	44
Figur 17: Utmatning under körning .....	45
Figur 18: Hydraulik detaljer under körning .....	46
Figur 19: Information om pappersrulle under körning .....	46

## Tabellförteckning

Tabell 1: Kommunikation mellan WinCC och PLC .....	8
Tabell 2: Tidsplan .....	11
Tabell 3: Ex Strukturtag .....	13
Tabell 4: Språkfiler .....	22
Tabell 5: Enhetsnummer .....	35
Tabell 6: Protokollstruktur brevlåda .....	36
Tabell 7: Lamb riktlinjer .....	42

# 1 Generell struktur

Grundstrukturen för denna rapport följer den ordning som vi utfört de olika arbetsmomenten i.

- Kapitel 2 innehåller rapportens inledning och en kort historik om Bravikens pappersbruk.
- Kapitel 3 beskriver Siemens WinCC, den utvecklingsmiljö vi utfört arbetet i.
- Kapitel 4 beskriver systemets syfte, den kravspecifikation vi kom fram till och de delar som ingick i vår lösning samt en tidsplanering av projektet. Varje sådan del beskrivs utförligt med bakgrund och hur vi utformat vår lösning. I de fall där det fanns flera tänkbara lösningar anges de olika lösningarnas för- och nackdelar och även vilken vi använde oss utav.
- Kapitel 5 beskriver hur vi testade vårt system. Både funktionalitet och prestanda testades.
- Kapitel 6 innehåller beskrivningar på hur vi löst de problem vi stött på under arbetets gång.
- Kapitel 7 innehåller vilka erfarenheter vi fått och vad vi lärt oss.
- Kapitel 8 innehåller de slutsatser vi dragit av detta projekt.

Rapporten avslutas med följande bilagor:

- Bilaga A innehåller de bilder och ritningar som vi använt oss av under arbetets gång. Dock finns inte projektets konstruktionsritningar med då dessa inte får publiceras av upphovsrättsliga skäl.
- Bilaga B innehåller en tabell över de maskinenheter som ingått i projektet samt deras enhets nummer.
- Bilaga C innehåller specifikationen över det protokoll som vi tagit fram för överföring av rullinformation mellan PLCn och WinCC. Detta beskrivs även i Kapitel 4.4.9.
- Bilaga D innehåller kravspecifikationen för systemet.
- Bilaga E innehåller de riktlinjer som Lamb AG har satt upp för utveckling av projekt i WinCC.
- Bilaga F innehåller bilder som visar hur det färdiga systemet ser ut under körning.
- Bilaga G innehåller en ordlista över de begrepp och termer vi använt oss av i rapporten. Dessa ord i rapporten har markerats med kursiv stil.
- Bilaga H innehåller den användarmanual som skrivits.

## 2 Inledning

Vi fick i uppdrag att konstruera grafiska terminaler med styrmöjligheter för magasinering av pappersrullar. Systemet byggdes åt Bravikens pappersbruk i Norrköping och den del som vi har utvecklat ska hantera Magasin 3, där rullarna sorteras och lagras. Bravikens pappersbruk består idag, de nya utbyggnaderna inkluderade, av 3 pappersmaskiner (PM51, PM52, PM53), 2 emballeringslinjer (WL1, WL2), 4 magasin med tillhörande sorteringsbord (WH1, WH2, WH3, WH4) samt ett transportsystem som binder samman alltihop. För en överblick av fabriken se Figur 11: Översikt Braviken i Bilaga A.1.

Anledningen till att man vill kunna studera fabriken grafiskt är för att man då kan övervaka hela fabriken från ett ställe. Detta system kommer inte att ersätta de gamla metoderna med manöverstationer utan kommer tillsammans med den gamla tekniken att underlätta för operatörerna. Detta genom att övervakningen kan ske från manöverrum med bättre arbetsmiljö än ute på fabriksgolvet. Man kommer i och med detta kunna övervaka och styra maskindelar utan att behöva vara på plats.

## **2.1 Information och historik om Bravikens pappersbruk**

Bravikens pappersbruk ligger strax utanför Norrköping. Bruket har tre pappersmaskiner och är väl känt för sitt vita och färgade papper som bland annat används för telefonkataloger, ekonomi- och sportpress. Braviken är idag ett av världens modernaste bruk för tidnings- och katalogpapper.

Huvuddelen av Bravikens produktion säljs på export och då framförallt till europeiska tidningar, endast 10 % stannar inom Sverige. Bravikens totala tillverkningskapacitet uppgår idag till 700 000 ton per år och har idag drygt 700 personer anställda.

Bravikens pappersbruk invigdes 1977 och brukets andra pappersmaskin stod klar 1985. Under perioden 1994-1996 genomfördes en omfattande utbyggnad av pappersbruket. Produktionskapaciteten utökades då från 440 000 ton till 700 000 ton per år. Utbyggnaden omfattade bland annat en pappersmaskin till PM53, som sedan starten haft ett flertal världsrekord i produktionshastighet. Det senaste världsrekordet är från juni 2000 och löd på 1811 m/min, vilket är det samma som 108,7 km/h. Tack vare denna höga produktionshastighet kan det ur varje maskin varje dygn rullas ut en pappersbana med en längd som motsvarar sträckan Braviken – Gibraltar.

Vintern 2001/2002 gjordes ännu en utbyggnad och det var i denna som vi var inblandade.

## **2.2 Rullhantering**

Detta avsnitt beskriver hur en pappersrulle tillverkas och vad som händer då den transporteras till lagret.

### **2.2.1 Rullhantering fram till magasinering**

Processen att tillverka papper kan delas upp i en mängd olika moment, allt från skogsavverkning till laminering, men här är vi bara intresserade av den del som återfinns från att papperet är färdigt till att det paketerats för leverans från pappersbruket.

Pappersmaskinen tar i ena änden in pappersmassa och bleker, blöter och torkar denna så att det i andra änden kommer ut färdigt papper. Detta papper kommer ut från pappersmaskinen på en så kallad pappersbana; papperet är då i storleksordningen 5-10 m brett och rullas kilometervis upp på stora valsar till stora pappersrullar. Dessa stora pappersrullar lyfts bort från pappersmaskinen till en rullmaskin. Denna maskin rullar av pappersbanan från valsen igen, matar det genom ett knivarrangemang som delar pappersbanan i smalare remsor som i



sin tur rullas upp till mindre pappersrullar. Vanligtvis så kapar man också papperet på längden så att det från en full pappersvals kommer två, tre omgångar med färdiga pappersrullar.

De pappersrullar som kommer ut från rullmaskinen har nu den längd, bredd och vikt som kunden beställt. Pappersrullarna är nu alltså i den storlek som till exempel ett tryckeri vill ha dem för att det ska passa deras tryckerilinje. En vanlig pappersrulles storlek kan vara 1-2 meter bred med en diameter på 0,7-1,2 meter. Denna standardrulle har vidare en vikt på någonstans mellan 0,5 och 1 ton.

För att pappersrullen ska anses klar för leverans krävs ännu ett steg, den så kallade emballeringen. Detta innebär att pappersrullen slås in som ett paket, vanligtvis i lite kraftigare omslagspapper eller kartong. Denna inslagning går till som så att pappersrullen först rullas in i omslagspapperet, varefter ändarna stängs till med cirkulära kartongskivor, så kallade rondeller. Efter att pappersrullen har emballerats så är den klar för leverans.

### **2.2.2 Rullhantering i magasinet**

Det är den här delen av fabriken som vi har varit inblandade i. Då en leverans oftast består av mer än en pappersrulle samt att pappersrullarna kommer ut blandade från de olika pappersmaskinerna behöver man sortera ihop dem innan de körs ner till lagret.

Det första som händer med pappersrullarna när de kommer in i magasin 3 är att de snurras 90 grader på ett vridbord. Efter detta transporteras pappersrullarna till rätt ramp på sorteringsbordet där de sorteras efter ordernummer. När ordern är samlad släpps pappersrullarna iväg mot lagret i samlad trupp. Då sorteringsbordet och lagret ligger på olika plan i byggnaden behöver pappersrullarna åka i en hiss innan de når slutstationen. Efter att pappersrullarna har åkt ner i hissen kommer de av rullresaren att ställas på varandra två och två. Detta för att det ska gå att stapla dem på lagret. Efter detta är pappersrullarna klara att hämtas med truck.

## 3 Förutsättningar och utvecklingsverktyg

### 3.1 Utvecklingsmiljö

Målplattformen för systemet var en standard PC med Windows 2000 som operativsystem. För att det inte senare skulle uppstå några oförutsägbara komplikationer vid installation och körning, användes en PC med operativsystemet Windows 2000 vid utvecklingen av systemet.

Systemet i sig är utvecklat i Siemens programvara WinCC 5.0 servicepack 2 som inriktar sig specifikt på utveckling av industriapplikationer. Förutom WinCC användes Siemens programvara Step7 5.1 servicepack 3 för PLC konfiguration och programmering. Eftersom detta system inte kommer att köras lokalt utan över ett internt nätverk har vi även fått använda oss utav en Ethernet förbindelse för att kunna uppnå en hög överföringshastighet mellan PLC maskinerna och de grafiska terminalerna som utvecklats i WinCC.

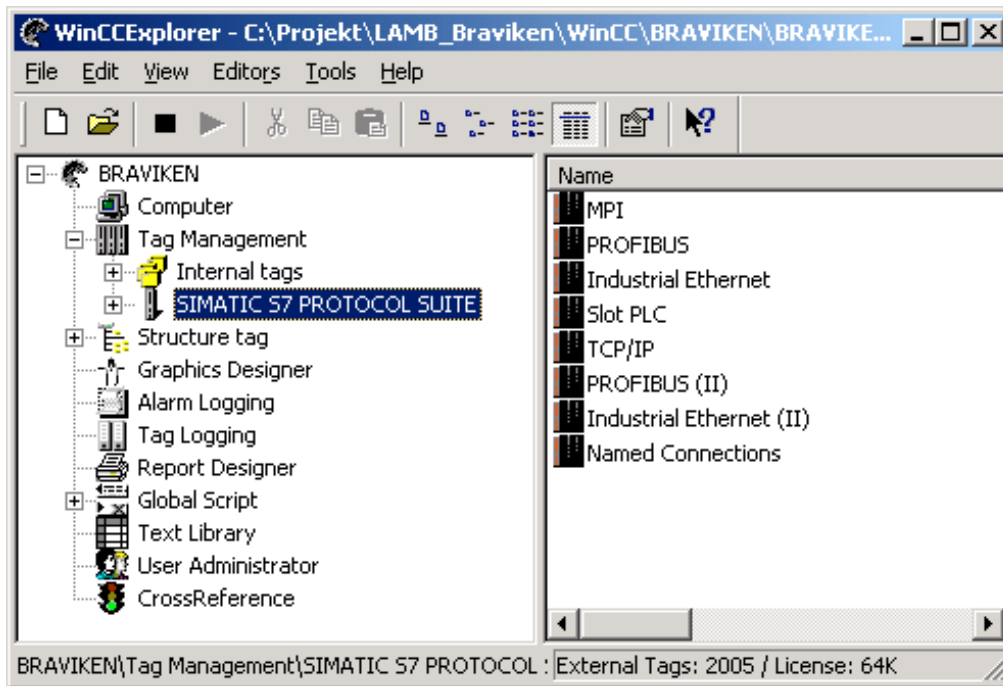
### 3.2 Beskrivning av WinCC

WinCC är en industri- och teknologineutral utvecklingsmiljö med inriktning mot visualiserings- och processkontrolleringsapplikationer inom produktions- och processautomationsmiljöer. WinCC innehåller moduler för grafisk visning, meddelanden, arkivering och rapportering av processdata. Dessa moduler har sedan anpassats speciellt för industriellt bruk. Speciell omtanke i utvecklingsmiljön har ägnats åt hög prestanda, snabb bilduppdatering och säker dataarkivering för att kunna erhålla hög tillgänglighet.

Förutom dessa systemfunktioner tillhandahåller WinCC öppna gränssnitt för användarlösningar. Detta gör det möjligt att integrera WinCC i komplexa automationslösningar. Data kan göras tillgänglig via *ODBC* och *SQL*. Förutom detta är dokumentlänkning via *OLE2.0* och *ActiveX-kontroller* möjligt. Detta gör det lätt att dela data mellan program i en Windows kompatibel miljö och då framförallt i en Windows NT miljö eftersom programmet är utvecklat för att köras där.

#### 3.2.1 Delprogram

För att starta WinCC väljs WinCC Explorer från Windows' startmeny. WinCC Explorer är det som binder samman och ger lätt tillgång till de olika program delarna som var och en stöder en viss del av Human Machine Interface-systemet, *HMI-systemet*.



Figur 1: WinCC Explorer overview

### 3.2.1.1 Graphics Designer

Graphics Designer är ett vektorbaserat ritprogram för att skapa processbilder. Med de många färdiga grafiska objekten som återfinns i objekt- och stilpaletterna kan man snabbt färdigställa även komplexa processbilder. *Actions* kan kopplas på en mängd olika sätt till varje individuellt grafiskt objekt. För att underlätta skapandet av *actions* finns guider som automatiskt genererar ofta förekommande dynamiska kopplingar.

### 3.2.1.2 Alarm Logging

Alarm Logging ger tillgång till visning och kontroll över insamling och arkivering av resultat. Man kan välja mellan följande typer av meddelandeattribut: block, klasser, typer, visning och rapportering. För att sedan visa meddelanden i processbilderna under körning använder man sig av Alarm Control objekt som man kan hitta i Graphics Designer.

### 3.2.1.3 Tag Logging

Tag Logging får processdata från exekverande processer och förbereder dessa för visning och arkivering. Dataformatet för arkiven och tidsperioder för datainsamling och arkivering kan väljas fritt. För att visa processvärden, används WinCC Online Trend och Table-kontrollerna, vilka visar data i trend- eller tabellformat.

#### 3.2.1.4 Report Designer

Report Designer är ett integrerat rapporteringssystem i WinCC för tids- och händelsestyrd dokumentering av meddelanden, operationer, arkivinhåll samt aktuell eller arkiverad processdata. Detta sker i form av användarrapporter eller projektdokumentation. Report designern används alltså för tillverkning av de olika rapporttyperna och har för detta en mängd olika standard layouter och utskriftsmallar.

#### 3.2.1.5 Global Script

I Global Script samlas alla de allmänna C funktioner som man skapar i projektet. Dessa funktioner kan sedan lätt även förflyttas mellan olika projekt. Dessa *C-script* används ofta för att konfigurera *actions* hos olika objekt, och kompileras av en i systemet inbyggd C kompilator. Global Script funktioner exekveras i körläge och deras exekvering startas av en *trigger* som då oftast är tagg som sätts utav en *PLC*.

#### 3.2.1.6 Text Library

I Text Library kan texter läggas in som sedan används av de andra modulerna. Här definieras även översättningar till andra språk samt vilka språk systemet ska klara av.

#### 3.2.1.7 User Administrator

User Administrator används för att tilldela och kontrollera användarnas åtkomsträttigheter till de olika editorerna för konfigurations- och körlägesystemen. Rättigheterna till WinCCs funktionalitet kan tilldelas per individuell användare. Upp till 999 olika auktorisationer kan allokeras, och även ändras i körläge.

#### 3.2.1.8 Cross Reference

Cross Reference används för att lokalisera och visa alla platser där ett objekt kopplats till en bild eller en funktion. Med länkningsfunktionen kan kopplingen sedan ändras utan risk för att inkonsistens införs i datamängden.

#### 3.2.1.9 Använda delar

De delar i WinCC vi behövt använda oss av i projektet var Graphics designer, Alarm logging, Global script samt Text library.

## 3.2.2 WinCC Basic Options

### 3.2.2.1 Client/Server

Med dess inbyggda klient/server funktionalitet kan WinCC användas i ett nätverk bestående av ett flertal koordinerade operations- och visningsstationer. Upp till 64 stationer kan ingå i ett enskilt projekt.

### 3.2.2.2 Redundancy

WinCC Redundancy erbjuder möjligheten att köra två sammanlänkade server PCs parallellt så de kan observera varandra. Om då en av dem skulle falla tar den andra över kontrollen av hela systemet. När den fallerande servern återställs till funktionsdugligt skick synkroniseras automatiskt alla meddelanden och processarkiv från den fungerande servern.

### 3.2.2.3 User Archives

User Archives är en databas som fritt kan konfigureras av användaren. I denna kan processdata lagras kontinuerligt på en server och visas online. Vidare kan kvittenser och börvärdestilldelningar för anslutna styrdon lagras.

### 3.2.2.4 Kommunikation mellan WinCC och PLC

Följande tabell innehåller de kommunikationssätt som stöds mellan WinCC och *Simatic S7*. Det finns även andra sätt att kommunicera med tredjepartstillverkare av PLCs. Vi har dock inte tagit med detta då det inte varit relevant för detta projekt.

<b>Protokoll som WinCC använder mot PLCn</b>	<b>Nätverks protokoll</b> (Olika hårdvaruförbindelser)
S7 protocol	<i>MPI Interface</i>
S7 protocol	<i>PROFIBUS</i>
S7 protocol	<i>Industrial Ethernet</i>
	<i>TCP/IP</i>
	Slot PLC
PMC communication	S7 PMC PROFIBUS

*Tabell 1: Kommunikation mellan WinCC och PLC*

### 3.2.2.5 Sammanfattning

Dessa delar speglar den inriktning som WinCC har mot automationslösningar inom en industriell miljö. För vårt arbete har denna inriktning visats sig vara mycket användbar och har verkligen underlättat arbetet.

Funktioner för felsäkerhet är något som behövs i sådana här system då man kommer att köra ett processkritiskt system på vanliga datorer. Detta då man önskar hög tillförlitlighet, pålitlighet och skalbarhet. Hög tillförlitlighet och pålitlighet uppnås genom att man kör flera servrar parallellt med redundant data. Skalbarhet uppnås genom att man kör flera servrar som belastningen sprids ut på.

I vårt projekt var inte skalbarhet något problem utan vi behövde bara hög tillförlitlighet och pålitlighet, varför vi använde oss av två stycken datorer med varsin installation av WinCC-serverprogramvaran. Dessa servrar sammankopplades för att uppnå redundant drift. Om den ena servern då skulle gå ner, kommer den andra servern att ta över driften. Detta resulterar i att användarna inte kommer märka av driftstörningar som bara påverkar den ena servern.

För att systemet ska gå att övervaka från flera skilda platser i fabriken har vi även använt oss av WinCCs inbyggda klient/server-funktionalitet. För en överblick av nätverkets uppbyggnad se bilaga A.4. Detta var även något vi behövde ta hänsyn till i konstruktionsarbetet, då systemet var tvunget att klara av mer än en klient samtidigt.

## **4 Utveckling av systemet**

### **4.1 Syfte**

Projektets syfte var att konstruera grafiska terminaler för övervakning av de pappersrullar och maskinenheter som förekommer i ett magasin på ett pappersbruk. Detta skulle utvecklas med hjälp av Siemens utvecklingsverktyg WinCC.

Vidare ska projektet ge oss en inblick i hur det är att arbeta med ett projekt med alla de moment som ingår och de problem som uppstår.

### **4.2 Omfattning**

Det projekt som pågår på Braviken omfattar en ny emballeringslinje, omformning av rulltransportsystemet mellan de olika pappersmaskinerna och magasinerna samt byggnation av magasin 3. Leverantör av detta är Lamb AG, som i sin tur har lagt ut elkonstruktion, byggnation och idrifttagning av den del som rör magasin 3 till underleverantören Wermlands Elautomation AB. Wermlands Elautomation AB har gett oss i uppdrag att utveckla det grafiska gränssnittet för den del som de ansvarar för.

Den del av projektet som vi ska utveckla är den grafiska övervakningen av den del av rullhanteringssystemet som rör sorteringsbordet i magasin 3, med tillhörande transportörer och övriga maskinenheter.

### **4.3 Projektplanering**

#### **4.3.1 Tidsplan**

Tabell 2: Tidsplan visar för varje moment den planerade tidsåtgången och den tid det sedan tog att utföra varje moment. De vita raderna visar den planerade tidsåtgången och de grå raderna visar hur arbetet förlöpte. Tabellhuvudet anger veckonummer med start vecka 49 år 2001.

Som synes i tabellen är det larmhanteringen som har blivit mest förskjutet. Detta beror framförallt på att underlag inte har funnits färdigt vilket har lett till att vi inte har haft något att jobba efter. Förutom detta har hela projektet i snitt blivit förskjutet en vecka, mest beroende

av att kraven ändrats för moment som redan var gjorda och som då fick göras om. Speciellt märks detta i vecka 8 då konstruktion av bakgrundsbilder har tillkommit. Detta berodde på att kunden gjorde ändringar i kravspecifikationen.

Moment Planerat, utfört	49	50	51	52	1	2	3	4	5	6	7	8	9	10	11	12	13
Projektmöten för utvecklingen					X	X											
					0	0											
Inläring av produktutvecklingsmiljö	X	X	X		X	X											
	0	0	0		0	0	0										
Kravspecifikation		X	X		X												
		0	0		0												
Konstruktion av bakgrundsbilder							X	X	X								
							0	0	0			0					
Konstruktion av dynamiska rullar								X	X	X							
							0	0	0								
Larmhantering										X							
													0	0			
Uppsättning av taggbibliotek								X	X								
								0	0	0	0						
Programtest och -verifikation											X	X	X				
											0	0	0				
Användarmanual													X	X	X	X	X
														0	0	0	
Rapportskrivning											X	X	X	X	X	X	X
										0		0	0	0	0	0	0

Tabell 2: Tidsplan

#### 4.3.2 Kravspecifikation

Kravspecifikationen har vi tagit fram genom att sammanställa de krav vi fått från slutkunden Bravikens pappersbruk, projektledaren på Lamb AG och vår uppdragsgivare Wermlands Elautomation.

Utöver de krav som de andra intressenterna ställde, lade vi själva till några krav för att testningen skulle gå att genomföra på ett bra sätt. Till dessa krav hör bland annat kravet D.3.2, att undersöka nätverkstrafiken.

En utförlig kravspecifikation med beskrivning på de delar som kom att ingå i systemet finns i bilaga D.



## 4.4 Beskrivning av konstruktionslösningen

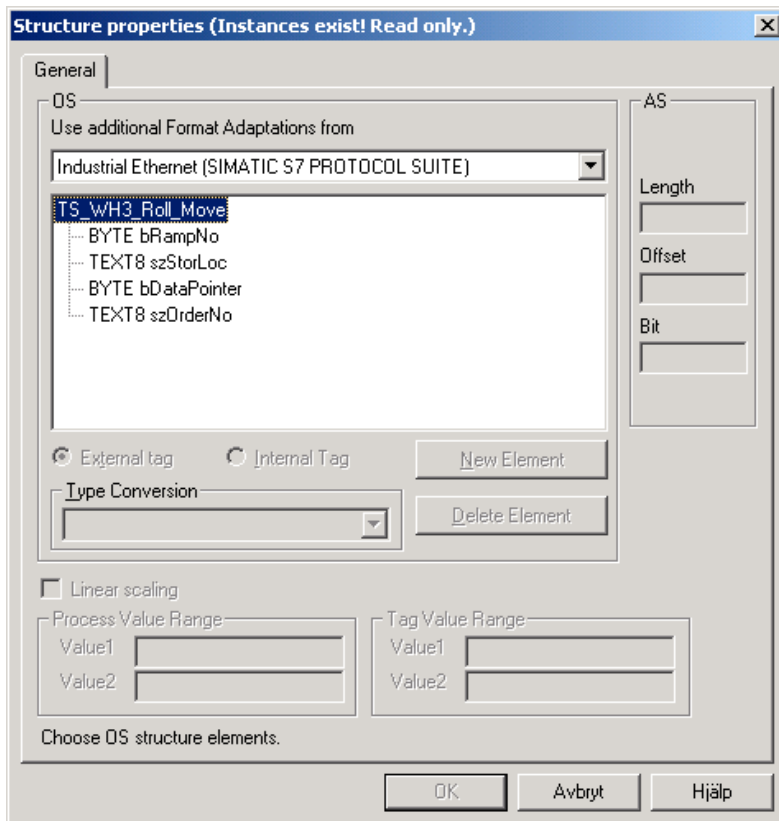
### 4.4.1 Tagbeskrivning

Taggar är det i WinCC som används som *variabler*. Dessa kan användas antingen som interna eller som externa taggar. En intern tag är en tag som hämtar och lagrar sitt innehåll internt i WinCC. Den har inte någon koppling mot omvärlden. En extern tag däremot är kopplat mot en yttre enhet som den hämtar sitt innehåll ifrån. Den externa enheten kan exempelvis vara en PLC.

WinCC är i sig den part som sköter all uppdatering av taggar vare sig de är interna eller externa. Detta kan ske på ett par olika sätt. Antingen sker det med jämna tidsintervall eller också sker det *upon change*, det vill säga då värdet till exempel har ändrat sig hos den externa enheten. Detta gör att man tillsammans med WinCC kan optimera dataflödet till och från taggarna. Indirekt behöver man alltså inte uppdatera och kontrollera en tag hela tiden om man vet att taggens innehåll bara kommer att ändras var 5:e sekund.

### 4.4.2 Uppsättning av taggar

De absolut viktigaste sakerna att hålla reda på hela tiden är de taggar man sätter upp. Det är nämligen dessa som är kopplingen till de dataområden som finns i PLCn där den aktuella informationen ligger. Taggarna sätter man oftast upp genom att man tidigare har skapat strukturer för just den maskindelstyp man håller på med. Ett exempel följer nedan.



Figur 2: Ex Strukturer

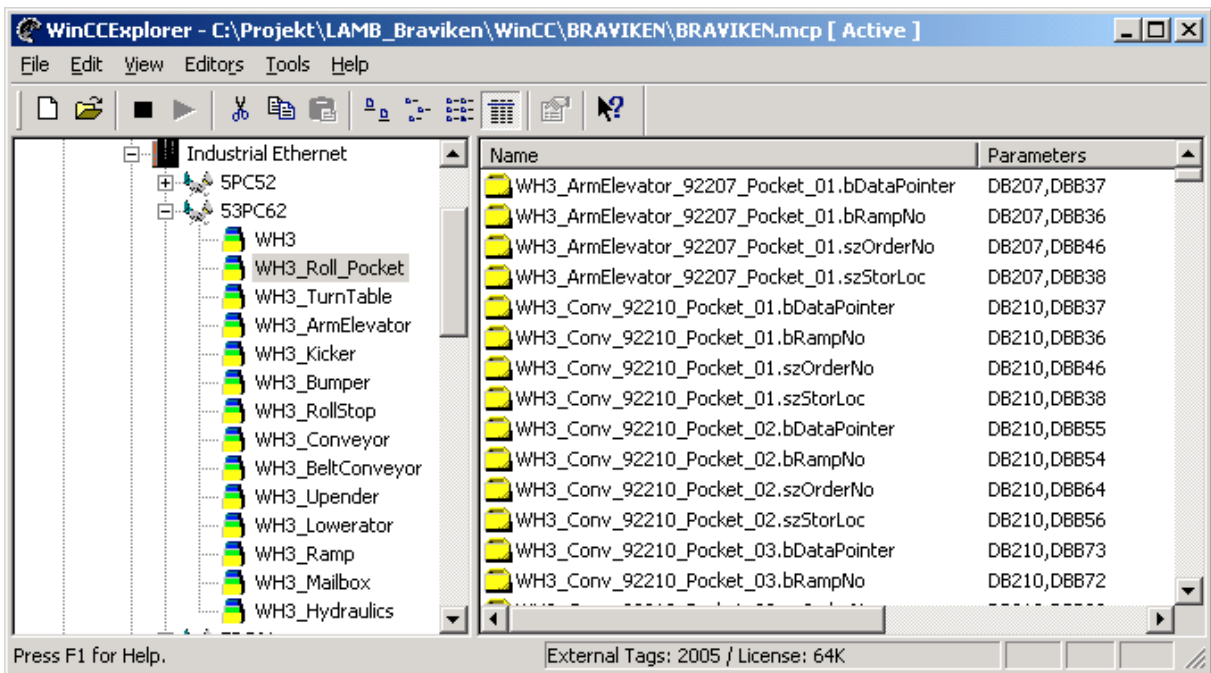
Här har vi skapat en struktur som vi använder oss av på transportören där mynt modellen används. Om hur mynt modellen fungerar kan läsas i kapitel 4.4.6. Här kan vi se att vi har skapat en struktur TS\_WH3\_Roll\_Move som innehåller följande.

Typ	Storlek	Offset	Namn	Beskrivning
BYTE	1	0	bRampNo	Rampnummret
BYTE	1	1	bDataPointer	Datapekaren
TEXT8	6	2	szStorLoc	En sträng som är 6tkn lång.
TEXT8	6	10	szOrderNo	En sträng som är 6tkn lång.

Tabell 3: Ex Strukturtag

Anledningen till att offseten för den sista strängen är två bytes förskjuten är att WinCC lägger utnyttjad längd på strängen i de första två byten. Detta gör att en sträng alltid tar upp exakt så mycket plats man anger att den ska plus de två extra byten för information om exakt hur lång den är.

När man sedan skapar instanser av strukturen fås följande.



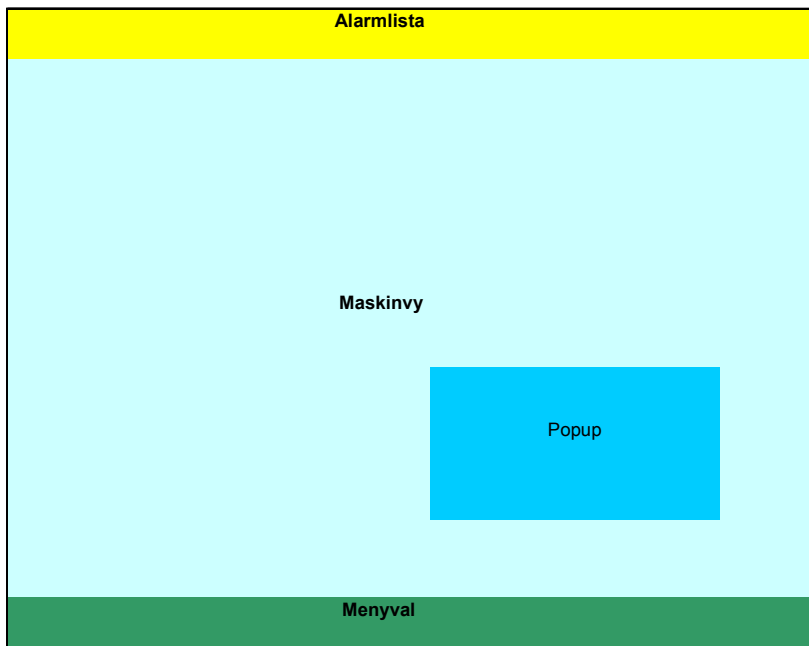
Figur 3: Ex Instans av struktur

Fördelen med att använda strukturer är framförallt tre. För det första kan man återanvända strukturer i senare projekt, det vill säga man kan kopiera dem och utföra ändringar för att anpassa strukturen för ett senare projekt. Den andra fördelen är att om man har många maskinenheter av samma typ behöver man bara tala om att de ska tillhöra en viss struktur så skapas alla individuella taggar automatiskt.

En annan fördel med strukturer är att de förenklar själva uppsättningen av taggar då man bara behöver ange vart strukturen ska börja. Detta gör att man inte behöver koppla alla taggar separat mot PLCns dataområden utan detta hanteras automatiskt med hjälp av offseten man satt i strukturen. Ett exempel är att för alla taggar som ska tillhöra enheten Armelevator 92207 har vi bara behövt ge ett namn till samlingen som i vårt fall blev WH3\_ArmElevator\_92207\_Pocket\_01. Sedan angav vi vilken struktur den skulle tillhöra samt vilket dataområde den skulle bindas mot i PLCn. I detta fall var det mot datablocket DB207 och på adressen DBB36 i detta datablock. För ett exempel på hur det ser ut i utvecklingsmiljön se Figur 3: Ex Instans av struktur.

#### 4.4.3 Terminalöversikt

Den bild som kommer att visas hela tiden systemet körs kallar vi för terminalbild. Den är uppbyggd av tre delar. Dessa tre delar ligger staplade vertikalt, enligt samma princip som en HTML-sida med frames.



*Figur 4: Terminalöversikt*

I den övre delen av Figur 4 finns en larmlista. Här kommer löpande nya larmer att visas. I den undre delen ligger menyn. I denna ligger knappar för språkbyte, utskrift av nuvarande vy samt snabbkopplingar till bilder över de andra delarna av systemet. Här finns även en knapp för att visa en större larmlista över alla de larmer som kommit.

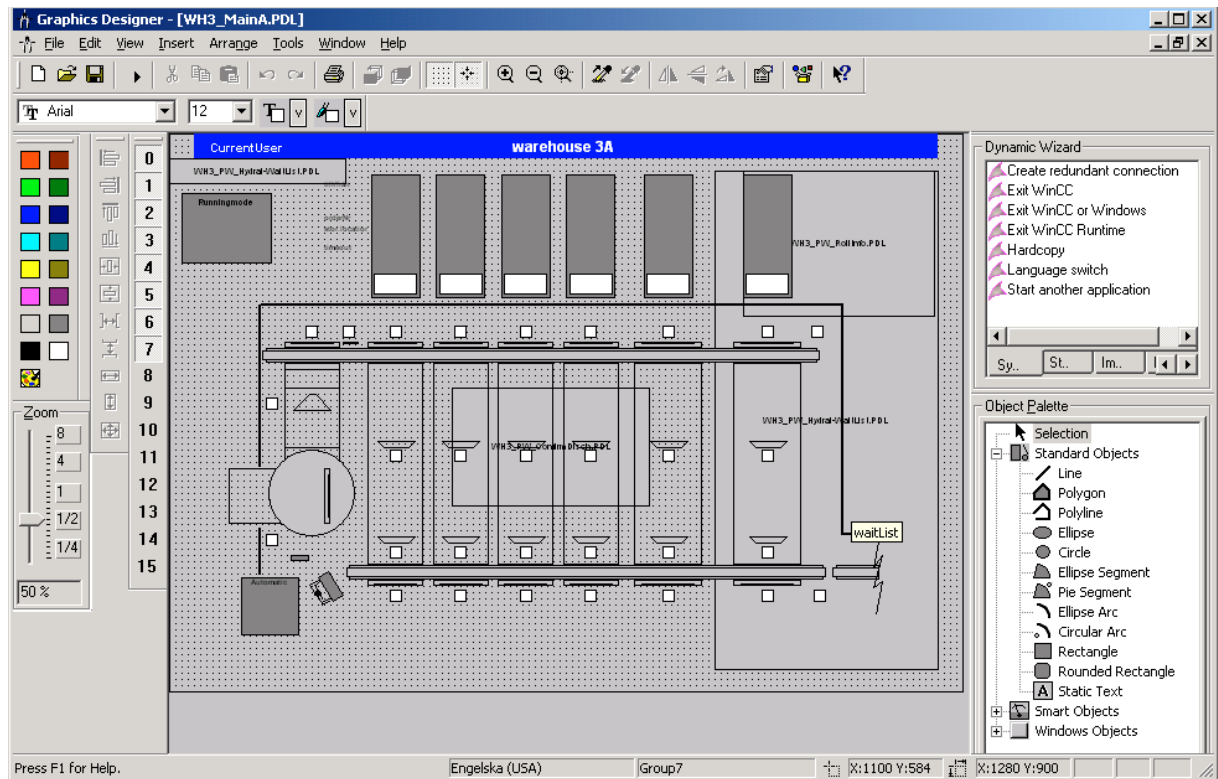
Den tredje, mittersta, delen av Figur 4 är den så kallade maskinvyn. Det är denna del som byts ut när man navigerar runt mellan de olika maskindelarna. Det är även här som den stora larmlistan kommer upp. Denna del kan visa en utzoomad bild där användaren kan välja vilken del av fabriken som är intressant att studera närmare. Se Figur 15: Översikt, startbild under körning. Denna bild är helt statisk och innehåller ingen information om systemets processtatus. Då användaren väljer ett visst fabriksområde ersätts innehållet i maskinvyn med en bild över det valda fabriksområdet. Denna nya bild är i sin tur uppbyggd av både statiska och dynamiska bitar som visar områdets processtatus vilket beskrivs närmare i de kommande två avsnitten.

Ovanpå maskinvyn kan det sedan öppnas små informationsfönster, så kallade popupfönster. Dessa kan flyttas runt och stängas som vanliga fönster i Windowsmiljön och innehåller sådan information som inte behöver visas löpande men som operatörerna ändå vill kunna se ibland.

Den del av fabriken som vi har utformat processbilder för delades upp i två olika bilder: en över sorteringsbordet och en över transportsystemet för pappersrullarna efter sorteringsbordet. För exempel på bilder av systemet under körning se bilaga F.

#### 4.4.4 Statiska bilder

För att visa vad som är statiska delar i en process bild kommer vi att använda oss utav en bild från övervåningen som innehåller ett sorteringsbord.



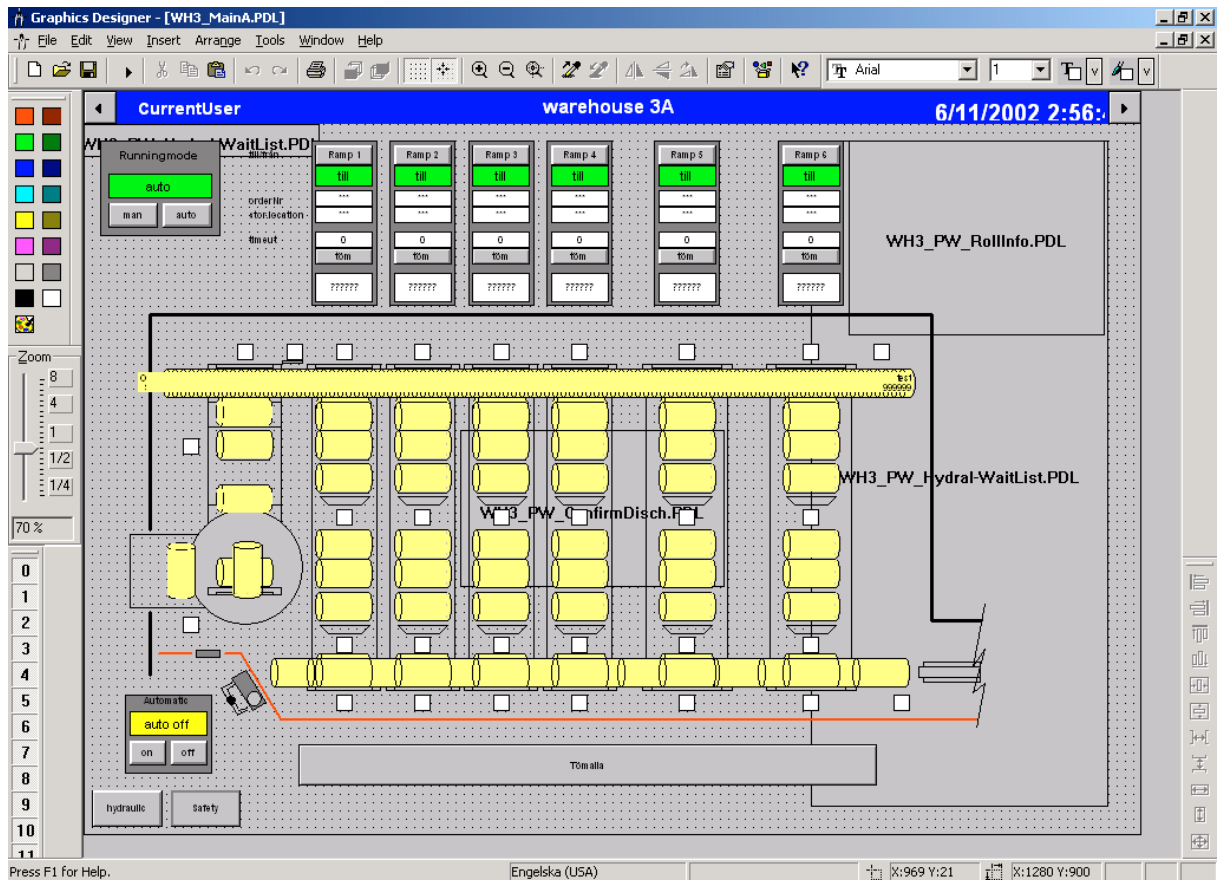
Figur 5: Ex Statisk bild

I den statiska delen av bilden ligger alla delar som inte kommer att ändras under körning. Dessa delar är framförallt fasta maskinenheter, såsom transportörer och ramper. Här ligger även de områden som är färgade för att på detta sätt rama in olika områden i bilden. Till exempel den mörka rutan längst upp till vänster ramar in den information och styrning som finns för detta körområde. Bilderna är uppbyggda enligt Lambs riktlinjer, vilket bland annat innebär att lager 0-7 innehåller statiska objekt. Allt om dessa riktlinjer finns i bilaga E.

För att framställa dessa bilder användes de ritningar som fanns över maskinlayouten. Dessa ritningar finns återgivna som Figur 12 och Figur 13 i bilaga A, Ritningar och bilder. Eftersom bakgrundsbilderna inte behövde vara så detaljerade som ritningarna är, använde vi bara ritningarna som mallar när vi ritade bilderna. När så själva maskinen var på plats placerade vi ut informationsfälten och knappar där vi ansåg att de var bäst placerade, till exempel så är rutorna med rampinformation placerade rakt ovanför respektive ramp.

#### 4.4.5 De dynamiska delarna i en bild

För att exemplifiera de dynamiska delarna i en bild använder vi även här en bild utav övervåningen.



Figur 6: Ex Statisk bild med dynamiska objekt

Alla objekt som nu tillkommit från den tidigare bilden Figur 5: Ex Statisk bild är exempel på dynamiska objekt. Det som skiljer dessa objekt från de statiska är att dessa på ett eller annat sätt kan ändras under körning. Ett par exempel på detta är alla pappersrullar, alla knappar och alla textfält vars egenskaper och attribut kan ändras under körning. Till de dynamiska objekten hör alltså allt som på ett eller annat sätt styr eller styrs av PLCn.

Eftersom bilderna följer Lambs riktlinjer, ligger alla dynamiska objekt i lagren 8 till 15. Dessa riktlinjer finns beskrivna i bilaga E.

#### 4.4.6 Pappersrullar med dynamisk storlek

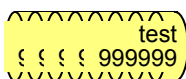
För att vi skulle få en så realistisk visning av pappersrullarna som möjligt bestämde vi oss i början av projektet för att använda oss av en modell så att vi kunde få en dynamisk storlek på pappersrullarna. Detta visade sig dock kräva både mer implementationsarbete av oss samt betydligt mer datorresurser än om alla pappersrullar hade visats med en fast storlek på

skärmen. När vi hade implementerat pappersrullar med dynamisk storlek på inmatningstransportören 92210 och kontrollerade mängden nätverkstrafik, såg vi att det inte skulle vara möjligt att använda denna lösning överallt. Detta visade sig dock inte vara något problem eftersom det, enligt kravspecifikationen, bara var på inmatningstransportören som det var viktigt med dynamisk storlek. Detta gjorde att vi använde fast storlek överallt förutom på inmatningstransportören 92210.

#### 4.4.6.1 Myntprincipen

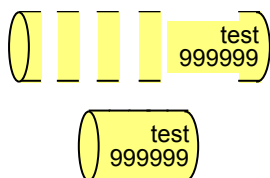
Den metod vi använde oss av för att lösa problemet med dynamisk storlek var att i stället för att ha fasta rullmodeller skapade vi små pappersrullar, så kallade ”mynt”. Vi satte sedan ihop ett antal sådana så att det såg ut som en stor pappersrulle. För att den stora pappersrullen som är ihopsatt av ett antal mynt skulle se ut som de vanliga fasta pappersrullarna var vi tvungna att ge mynten vissa egenskaper. Dessa egenskaper var att man skulle kunna välja att visa enbart bakkant, mittdel, framdel, textdel eller alltihop i någon kombination.

Så som mynten ligger packade på transportören i den slutliga lösningen visas i Figur 7.



Figur 7: Dynamisk storlek på rullar

För att visa hur allt fungerar på ett tydligare sätt har vi i Figur 8 dragit isär 5 mynt och visar sedan hur det ser ut när de sätts ihop för att forma en stor pappersrulle.



Figur 8: Sprängskiss av mynt

På den övre delen av Figur 8 ligger mynten brett isär för att visa hur vi kan välja att visa bakkant, mittdel, framkant samt textdel på mynten i rätt kombination. På den nedre delen har vi sedan flyttat ihop mynten igen så som de ska ligga. Då ser man hur vi har fått en hel pappersrulle med dynamisk inställd storlek.

För att myntmodellen skulle fungera var vi tvungna att ha väldigt många positioner på transportören som var och en kunde tala om fall det var en pappersrulle där eller inte. Närmare bestämt hade vi ett avstånd på 0,25 m mellan varje position på transportören vilket gav att för varje mynt som visas så är pappersrullen minst 0,25 m bred. Detta ger att en pappersrulle som är 1,25 m bred i verkligheten kommer att visas i form av 5 mynt.

#### 4.4.6.2 Implementering av myntprincipen

För varje myntposition finns ett dataområde i PLCn. Detta dataområde innehåller dels de texter som ska visas samt en datapekare. Denna datapekare fyller två funktioner. Den första funktionen är att den talar om fall det befinner sig en pappersrulle på aktuell position. Detta görs genom att kontrollera ifall datapekaren är noll eller inte. Om datapekaren är noll finns det ingen pappersrulle på den aktuella positionen. Om pekare är skild från noll så innebär det att det finns en pappersrulle där. Den andra funktionen är att om datapekarens innehåll är skilt från noll så är innehållet en pekare till det område i PLCn där den fullständiga informationen finns om pappersrullen.

Vi har sedan kopplat taggar i WinCC mot dessa dataområden i PLCn. Med hjälp av datapekaren kan man nu avgöra vilka delar av myntet som ska visas. Detta styrs på tre olika sätt. Om aktuell datapekare är skild från noll ska myntets mittdel visas. Om dessutom datapekaren för myntpositionen bakom är noll ska även myntets bakkant visas. På samma sätt fungerar det för framkanten som styrs av den aktuella datapekaren och datapekaren för myntpositionen framför. Om framkanten ska visas så ska även myntets textfält visas och uppdateras med aktuella texter från dataområdet i PLCn. I annat fall ska ingen del av myntet visas.

#### 4.4.7 Pappersrullar med fast storlek

För de transportörer där myntmodellen inte används har vi istället använt pappersrullar med fast storlek. Förutom en transportör var detta en godtagbar lösning och den enklare lösningen gav besparingar i resursutnyttjande både i WinCC och på nätverket. Den storlek dessa pappersrullar gavs är vald för att matcha en normalstor pappersrulle utan att vara för bred för de fall då många smala pappersrullar befinner sig på samma transportör.

Transportörerna är uppdelade i positioner precis som i myntmodellen, men istället för att ha positionerna lika tätt så ligger de med ett visst avstånd mellan sig. Detta gör att pappersrullarnas position inte kommer att visas exakt som i verkligheten men avståndet är valt så att det inte ska vara allt för stor felvisning. Eftersom positionerna ligger en bit isär kommer rullarna inte att flyta fram lika smidigt som då myntmodellen används utan de kommer istället att göra större förflyttningar åt gången, vilket ger en lite hoppig framfart. Alla positioner som en pappersrulle med fast storlek kan finna sig på är kopplade till ett eget dataområde i PLCn.

Visningen av pappersrullar med fast storlek kommer enbart att bero på den aktuella datapekarens innehåll, eftersom de bara kan visas på ett sätt. Om pekaren är skild från 0

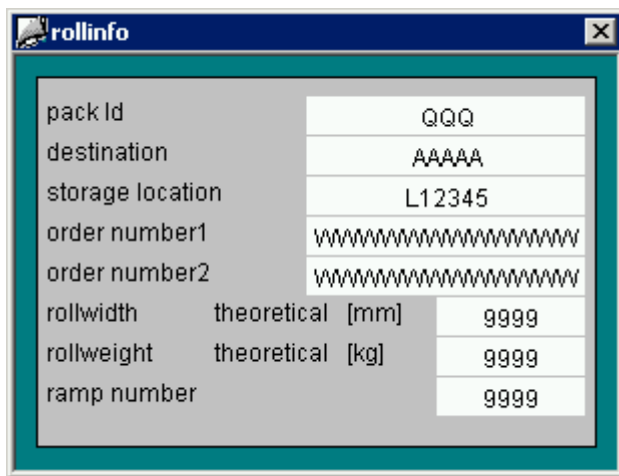


kommer rullen att visas och texterna att uppdateras med korrekt information från motsvarande dataområde. Om datapekaren är lika med noll, kommer rullen inte att visas alls.

#### 4.4.8 Allmänt för samtliga pappersrullar

Något som är gemensamt för alla pappersrullar vare sig de är av myntmodellen eller har en fast storlek är något som kallas för en *Mouse Action*. Denna fungerar som så att när någon trycker med vänster musknapp på en pappersrulle körs ett internt *C-script* igång. Detta *C-script* som hör till pappersrullen tar den aktuella datapekaren för pappersrullen och skickar ner dess innehåll till PLCn. PLCn svarar då med att kopiera all information om pappersrullen från rullens dataområde till ett speciellt dataområde. Detta dataområde kallar vi för brevlådan. För information om hur brevlådan och dess protokoll fungerar i detalj se under rubrik 4.4.9 Protokoll för visning av rullinformation och i bilaga C.1.

Efter att datapekaren har skickats ner öppnas ett popupfönster som ser ut på följande sätt, se Figur 9: Rullinfo. Detta popupfönster kommer att visa all den information som finns tillgänglig att hämta i brevlådan. När fönstret först kommer upp är innehållet i de dynamiska fälten tomma. De dynamiska fälten i Figur 9: Rullinfo är de som har en vit bakgrund.



Figur 9: Rullinformation

Detta fönster innehåller i sig det *C-script* som ser till att läsa upp informationen ur brevlådan samt uppdatera popupfönstrets textfält. Detta kommer dock inte att göras förrän innehållet i fältet för datapekaren som finns i brevlådan har samma innehåll som aktuell rullens datapekare. Detta gör att informationen blir fryst i popupfönstret. I och med detta kan samma brevlåda nu användas av flera klienter, vilket vi beskriver närmare nedan under rubriken Protokoll för visning av rullinformation.

#### **4.4.9 Protokoll för visning av rullinformation**

Problemet med att flera klienter kan komma att ha samma bild uppe och begära rullinformation genom att klicka på en rulle behövdes lösas på något sätt.

Ett alternativ var att ha ett visst antal brevlådor, där varje klient hade sin egen att läsa ur. Men detta alternativ skulle leda till att vi dels skulle vara tvungna att på något sätt meddela PLCn vilken klient som efterfrågade informationen och dels att varje klient skulle kräva en egen specifik konfiguration av brevlådan. Detta alternativ förkastades direkt eftersom det skulle ge ökad risk för fel med tanke på att programmet skulle bli mindre generellt på grund av att det skulle kräva konfigurering för varje enskild klient vilket då skulle ge ett inflexibelt system.

Ett bättre alternativ som vi kom fram till var att specificera ett protokoll för informationsutbytet, i vilket PLCn meddelar klienten när informationen är klar att hämtas och klienten då bara tömmer brevlådan en gång. Protokollet gör att alla klienter kan dela på en brevlåda, eftersom vi antar att det inte kommer att ske läsning av två klienter samtidigt. Med detta menar vi att hämtning inte kommer att ske samtidigt av två klienter under samma PLC-cykel, alltså inte under samma klockperiod hos CPU:n. Detta antagande bygger dels på att brevlådan inte kommer att användas om inte en operatör initierar informationshämtningen (genom att begära rullinformation) och dels på att det i praktiken inte har att röra sig om fler än 5 klientstationer.

För en djupare specifikation av det protokoll som vi skapade se bilaga C.1.

#### **4.4.10 Byte av språk**

Byte av språk var ett utav de krav som Lamb hade gett oss. Vi var därför tvungna att se till att systemet skulle gå att köra både i svenskt och engelskt läge. För att byta språk i systemet under körning finns två knappar längst ner till vänster i bilderna. Dessa knappar representerar varsitt språk. I vårt fall innebar detta svenska och engelska. Då man byter språk i systemet så sker två saker. Vi ställer dels om den inbyggda språkhanteraren till rätt språk samt kör igång en egen extern språkhanterare.

Om man bara hade använt den i WinCC inbyggda språkväljaren måste översättningen av knapptexter och dylikt ske i Graphic Designer. De olika språkens texter kan då inte ses samtidigt utan man ser ett språk i taget och måste utföra ett menyval för att se det andra språket.

Då denna lösning inte upplevdes som lättarbetad använde vi istället en alternativ lösning. Denna lösning går ut på att alla texter visas dynamiskt och är bundna till varsin Tag som

innehåller själva texten. Dessa Taggar fylls med lämpliga texter av ett *C-script* som läser texterna från textfiler där översättningarna finns.

Vilka textfiler som ska läsas in avgörs av en parameter till *C-scriptet*. Beroende av parametern bestämmer *C-scriptet* filnamnens slut och får på sätt tag på de korrekta texterna.

Vi skrev applikationen på svenska och engelska. De textfiler vi då fick var:

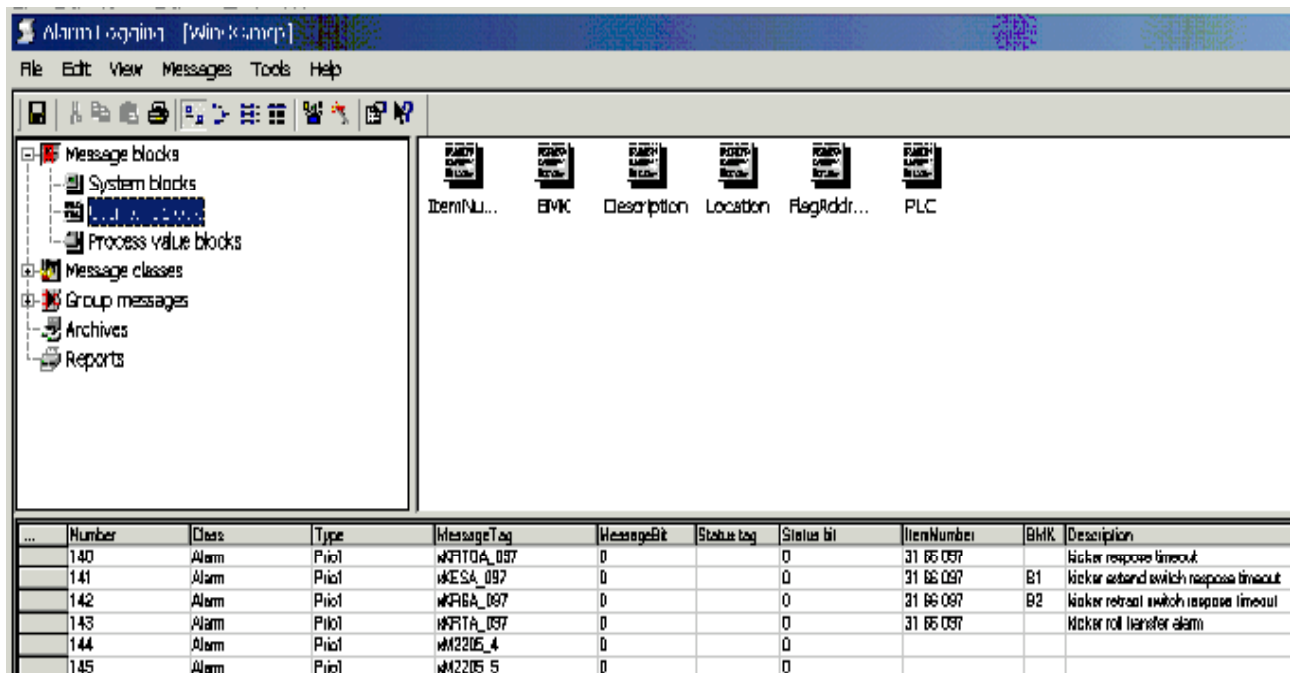
Filnamn	Innehåll
StaTxt10_ENG.txt	Statiska generella texter på engelska
StaTxt11_ENG.txt	Statiska projektspecifika texter på engelska
Button10_ENG.txt	Generella knapp texter på engelska
Button11_ENG.txt	Projektspecifika knapp texter på engelska
StaTxt10_SWE.txt	Statiska generella texter på svenska
StaTxt11_SWE.txt	Statiska projektspecifika texter på svenska
Button10_SWE.txt	Generella knapp texter på svenska
Button11_SWE.txt	Projektspecifika knapp texter på svenska

*Tabell 4: Språkfiler*

Med hjälp av den lösning vi valde blev det enklare att redigera de texter som finns på flera språk. Tyvärr kunde denna lösning bara appliceras på de statiska texter som finns i bilden och på texter till knappar. Hanteringen av larmtexter beskrivs i kapitel 4.4.11 .

Vi upptäckte också att denna lösning gjorde att vi kunde uppdatera och kontrollera texternas visning under körning. Detta genom att helt enkelt byta språk under körning vilket gjorde att texterna lästes in igen. Detta var något som var användbart då det var smidigt att kunna se hur texterna såg ut i körläge och man kunde på ett enkelt sätt testa olika alternativa texter för de båda språken.

## 4.4.11 Larmhantering



Figur 10: Alarm Logging editorn

I Alarm Logging definieras och konfigureras de larm som applikationen ska innehålla. Detta görs på så vis att man definierar de olika grader av larm som ska finnas, i vårt fall varning, fel och larm. Förutom detta sätts även vilka olika attribut man vill kunna visa för varje larm. Två exempel på attribut vi skulle visa i detta projekt var att för alla larm skulle det visas vilken PLC som rapporterat larmet samt vilken maskinenhet som larmet kommit från. För en fullständig lista över de attribut som skulle visas se bilaga D.3.1.1 under punkten Larmvisning.

För varje larm sätts sedan graden på larmet det vill säga om det ska vara larm, fel eller varning. I vårt fall är larm viktigast och varning minst viktig. Detta markeras med vilken bakgrundsfärg larmet visas i samband med. Förutom detta knyts varje larm till en Tag som anger statusen på larmet. I praktiken talar denna om ifall larmet och dess attribut ska visas i larmlistan.

Det är i Alarm Logging editorn som man matar in alla attribut för varje larm. Alla attribut matas in på det förinställda språket, vanligtvis engelska. Alla dessa olika texter måste man sedan översätta till övriga språk som systemet ska klara av att visa. Denna översättning görs i Text Library editorn. Under körning kommer sedan den inbyggda språkväljaren i WinCC att visa korrekta texter för de larm som har gått.

## 5 Test av systemet

Eftersom utvecklingen av styrsystemet som ska köras i PLCn har pågått under tiden vi har arbetat med det här projektet, har vi fått testa lite allt eftersom det blivit möjligt. På grund av detta har vi inte kunnat ha någon specifik testperiod då systemet skulle köras igång så fort som PLC-programmet var klart. Vi har därför fått testa våra lösningar i mindre skala och sedan när vi fått det att fungera använt de metoderna i hela projektet. I det stora hela har testningen utförts genom att när en viss del av PLC-programmet var klar, testade vi i WinCC att denna del fungerade korrekt.

Förutom att systemet ska vara korrekt fungerande hade vi även vissa prestandakrav att klara av. Eftersom vi fick ett testprogram till PLCn var detta något vi kunde testa relativt enkelt genom att helt enkelt mäta trafiken som uppstod på nätverket. För att kontrollera att mängden trafik på nätverket inte var för hög gjordes en mätning när alla pappersrullar var på plats. Mätningen gav att den totala trafiken på nätverket var strax under 3kbyte/s. Detta värde låg under vårt riktvärde på 10kbyte/s. Mätningen gjordes med den inbyggda nättrafikövervakaren i Windows 2000.

Den slutliga testningen och buggkontrollen skedde under slutet av systemets idrifttagning. Detta gick bra eftersom vårt system inte behövdes för att styra PLCn då detta även kunde göras via fasta manöverstationer.

## 6 Problem under konstruktion

### 6.1 Installation och inläring

Något som orsakade en hel del problem i inläringen var att vi inte hade någon licens till utvecklingsverktyget WinCC vilket ledde till att det stängde av sig ibland. Vi hade inte heller den senaste versionen vilket också orsakade en hel del systemkrascher. Detta visade sig senare berott framförallt på att vi hade en dator med för lite kapacitet samt fel servicepack för WinCC. Men efter att vi fått den nya datorn med de senaste servicepackarna till Windows 2000 och fått den senaste versionen av utvecklingsverktyget började allt rulla på och vi blev mer och mer införstådda i hur WinCC fungerade.

Något som gjorde att en del problem upplevdes som värre än de egentligen var, var att vi inte hade någon person att fråga om hjälp förutom vår kontakt hos Lamb AG i Schweiz. För att sedan göra saken ännu sämre var inte Siemens de bästa på att skriva dokumentation och hjälpfiler till sina programvaror. Manualen som fanns var helt enkelt katastrofalt dålig. Förutom detta var deras hemsida lika dålig, så den var inte mycket att ha heller. Vi fick gå den hårda vägen hela tiden, det vill säga vi fick helt enkelt testa allt vi ville ha svar på. Ett typiskt exempel på detta problem var onlinehjälp för de funktioner som fanns tillgängliga i *C-scripten*. Denna hjälp hade direktlänkar till kodexempel i en annan hjälpfil. Men när man följde en direktlänk kom man inte till kodexemplet som man hade förväntat sig, utan istället öppnades hjälpfilen på en tom sida där man sedan fick upprepa sökningen, för att först då komma fram till den fungerande länken till kodexemplet.

### 6.2 Uppsättning av nätförbindelse

Vid uppsättningen av nätförbindelsen mellan datorn med WinCC och PLCn var det hårdvarukonfigureringen som gav oss mest problem. Något vi lärde oss ganska snabbt var att kontrollera så att den senaste versionen av Siemens paketen fanns installerat på båda sidor av förbindelsen. Detta då Siemens, som utvecklat miljön, inte tänkt på att program bör vara kompatibla bakåt. Med andra ord, program ska fungera med äldre versioner av programvaran. Detta var ju något som vi inte tänkte på eftersom man har för vana att ett nyare program ska

kunna klara av en äldre versions funktioner. Ett exempel på detta är Microsofts Word 2000 som klarar av att läsa och redigera filer av typen Microsoft Word 97.

Dock efter att vi hade installerat uppdaterade kommunikationspaketet samt gjort rätt konfigurationsinställningar både i PLCn och i PCn fick vi nätverket att fungera som vi önskade. För en överblick av nätverket se bilaga A.4.

## 6.3 Övriga problem

### 6.3.1 Projektmiljö

Ett av de största problemen vi egentligen stötte på var att vi aldrig tidigare hade jobbat eller kommit i kontakt med något liknande arbetsområde. Detta problem löstes dock bra genom att vi åkte på ett studiebesök till pappersbruket Stora Enso Gruvön. Där fick vi en guidad tur genom maskinhallen samt blev visade ett Siemens WinCC system som nyss blivit installerat och igångkört. Visserligen gällde systemet vi fick se inte ett sorteringsbord utan en emballeringslinje, men vi fick ändå en viss bild av hur det hela skulle hänga ihop. Vi fick även ut av besöket hur pappersmaskinens delar såg ut och hur de fungerade i verkligheten. Denna överblick hade vi nytta av då vi skulle visualisera de olika maskindelarna på skärmen.

### 6.3.2 Utvecklingsverktyg

Ett annat problem vi stötte på var att vi aldrig tidigare hade använt utvecklingsverktyget WinCC. Detta gjorde att vi kände oss ganska begränsade i början men efter att vi haft en intensiv inlärningsperiod då vi satt och testade olika typer av lösningar och alternativ kom vi fram till vad som var möjligt att genomföra och vad verktyget var mindre lämpat för.

Som vi tidigare har beskrivit var det möjligt att göra *C-script* i verktyget. Detta var dock väldigt olämpligt att använda i vissa situationer och då framför allt när man ville ändra egenskaper på många olika objekt samtidigt. Gjorde man detta med hjälp av ett *C-script* utfördes nämligen alla ändringar separat och detta ledde till att systemet kändes trögt och ryckigt då de dynamiska delarna av bilderna uppdaterades vid olika tidpunkter. Det tydligaste exemplet på detta var när pappersrullarna flyttades så hängde inte texterna med utan de uppdaterades och visades efteråt. Detta fick vi då lösa på ett mer omständigt sätt.

Om man istället för att använda *C-script* för att ändra ett objekts egenskaper knöt ett objekts egenskap direkt mot en tag så buffrade WinCC denna ändring och utförde sedan de buffrade ändringarna samtidigt. Detta var något som WinCC skötte internt och som man som programutvecklare inte kunde styra över. Efter att vi gjort på detta sätt istället upplevdes

bilduppdateringarna mer synkroniserade och systemet kändes i och med detta mindre ryckigt även om det i sig inte blev något snabbare.

Något som den slutgiltiga lösningen ledde till var att vi inte kunde ha pappersrullarna som ett enda objekt utan var tvungna att dela upp dem i två mindre. Dessa delar var dels själva pappersrullen och dels ett objekt som innehöll de textfält som skulle visas på pappersrullen. I den första lösningen med *C-script* behövde vi bara ändra på ett ställe för varje position med pappersrulle. *C-scriptet* utförde nämligen i denna lösning alla de uppdateringar av de egenskaper vi behövde ändra på hos objektet. Men i den slutgiltiga lösningen när alla egenskaper var bundna direkt på de olika ställena i objektet blev vi tvungna att ändra var och en av dessa separat. Detta gjorde att vi nu istället för att vi skulle ha behövt ändra på ett ställe som vi gjorde när *C-script* användes blev vi tvungna att istället ändra på 11 ställen för varje pappersrulle med tillhörande text. I och med detta krävdes betydligt mer jobb för att utföra uppsättningen av alla pappersrullars egenskaper då vi hade att göra med cirka 180 olika positioner för pappersrullar.

Något annat som fungerade lite underligt i WinCC var att de taggar man hade satt upp inte gick att sortera efter adressordning. Istället kunde man sortera efter namn, datatyp eller adress. Fast när man sorterade efter adress följde de inte från minsta adress till den största utan sorteras först efter datatyp och sedan på adressnummer. Vad vi menar med detta är att alla taggar som är av typen byte kommer att sorteras efter varandra i stigande eller fallande adressordning. Detta gjorde att det inte gick att sortera taggarna strukturvis i adressföljd. Om det hade kunnat gå att sortera taggarna strukturvis och i adressföljd hade det blivit mycket enklare att leta reda på den tag man för tillfället skulle ha tag på då vi hade runt 5000 taggar uppsatta.

En annan sak vi har reflekterat över är att det hade varit bra om man hade kunnat ha en strukturtaggar i en annan strukturtag. Detta skulle ge att strukturer kunde ärva andra strukturers uppbyggnad. Om detta hade varit möjligt skulle man kunnat göra mer generella strukturer som man sedan hade kunnat återanvända när de riktiga strukturerna skapades.

En funktion som vi ansåg saknades i WinCC var att man skulle kunna använda buffring av objekt uppdateringar även i *C-script*. Detta för att det skulle fungera på samma sätt som för direktknutna egenskaper, det vill säga att tidpunkten för uppdateringen av alla ändringar som skulle utföras kunde styras.



## 7 Erfarenheter

Den viktigaste erfarenheten vi lärt oss av detta examensarbete är att vi har jobbat i ett riktigt projekt där tidspress, planering, samarbete och kommunikation varit en viktig del av den vardag vi jobbat i. Den tidsplan vi satte upp i början av projektet visade sig vara en bra bild av hur arbetet kom att utföras. Att vi valde det här projektet som examensarbete är absolut ingenting vi ångrar och att det varit ett riktigt projekt där slutprodukten skulle användas i fabriken drif gjorde det hela ännu mera intressant.

Vi har under tidens gång fått många nya lärdomar. Bland annat har vi lärt oss Siemens utvecklingsverktyg WinCC väldigt bra. Förutom detta har vi, även om vi inte har varit ansvariga för PLC-programmeringen, lärt oss mycket om hur man programmerar en PLC i Siemens Step 7 samt hur man konfigurerar den. Detta gav oss även en lärdom i skillnaden mellan hur man programmerar en PLC och hur man programmerar på en vanlig dator. Även om båda miljöerna är moderna samt att datorerna har samma uppbyggnad används assemblerliknande språk och ladderdiagram vid PLC-programmering. Program som ska köras på en vanlig dator är däremot oftast skrivna i ett högnivåspråk, som till exempel C/C++. Vi har genom detta projekt fått prova på att integrera dessa båda sätt, då vi utvecklat mindre PLC-program för att kunna testa och utvärdera de lösningar vi haft.

Projektets har också givit oss en inblick i hur det är att jobba i en projektgrupp med tidspress och krav på att uppgiften var tvungen att lösas innan deadline nåtts. Detta kändes väldigt svårt att förstå i början men har under projektets gång lett till att man har förstått hur det är att jobba tillsammans i ett projekt som dessutom var uppdelat över hela Europa. Förutom detta har det hela tiden tvingat oss att hämta in information själva och försöka förstå och hitta på egna lösningar, eftersom den hjälp som funnits att få befunnit sig i Schweiz större delen av projektets gång.

## 8 Slutsatser

Projektarbetet har inte medfört några större komplikationer då vi började i god tid. Det som gjort att utvecklingen stannat av har antingen berott på att vi väntat på ytterligare underlag eller att det tillkommit eller ändrats något i kravspecifikationen. Applikationen som utvecklats följer den kravspecifikation som finns i bilaga D. Vi behövde inte välja bort något område på grund av tidsbrist eftersom tidsplanen har fungerat väl.

För projekt av den här typen är WinCC ett lämpligt utvecklingsverktyg även om det har vissa svagheter. Bara att försöka utveckla ett liknande system i ett verktyg som inte är anpassat för den här typen av applikationer skulle kräva betydligt mer tid. Hade vi till exempel implementerat projektet i C/C++ hade vi behövt skapa en infrastruktur liknande den som redan finns i WinCC.

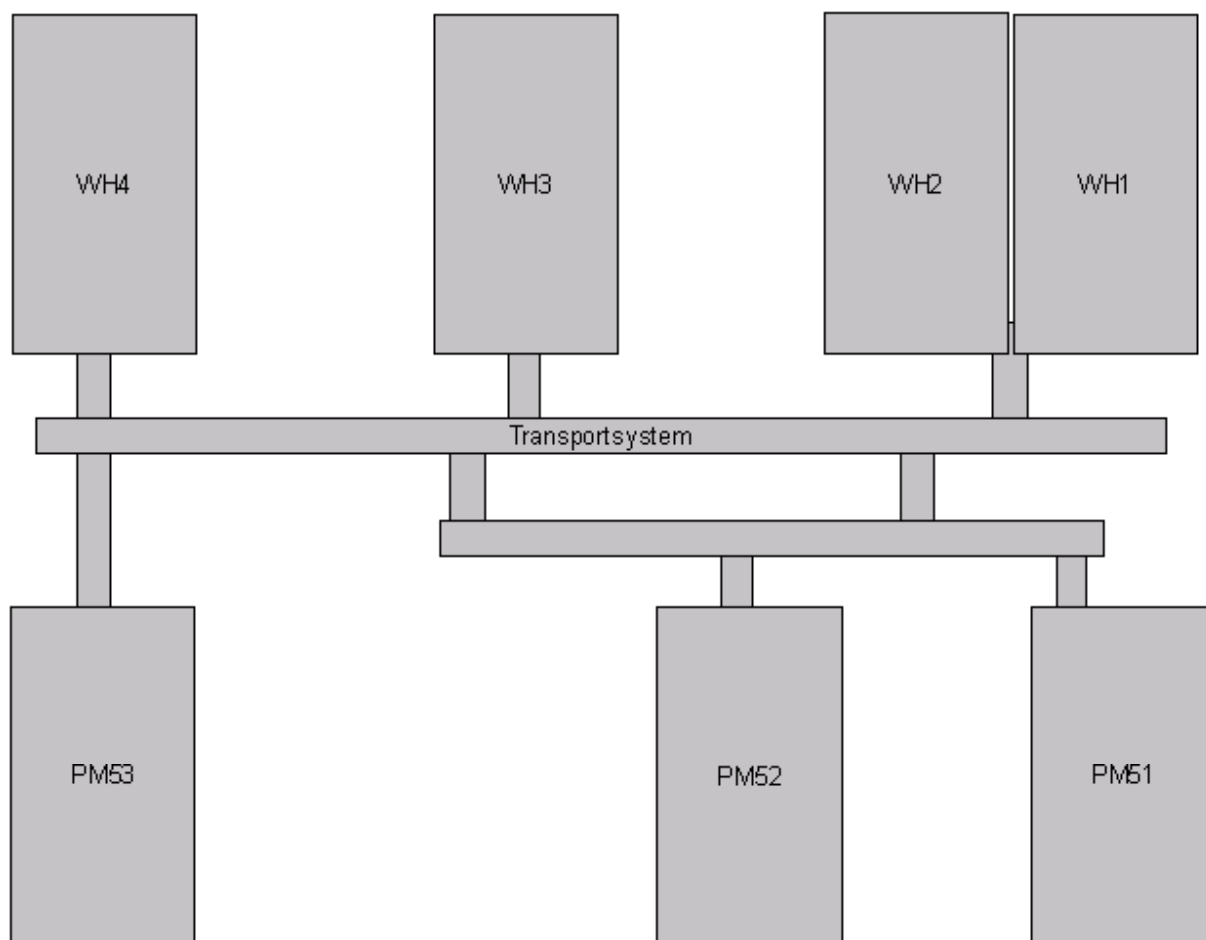
Som slutord har de mål och föresatser som funnits fungerat bra och uppnåtts. Samarbetet med handledarna både i skolan och på företagen har fungerat bra och det känns som om vi fått en uppfattning av hur ett projekt fungerar i verkligheten.

## Referenser

- [1] *Holmenpaper*, *Holmenpapers hemsida*, <http://www.holmenpaper.com/>, 2002
- [2] *Siemens*, *WinCC*, <http://www.siemens.com/wincc/>, 2002
- [3] *The C++ Resources Network*, *cplusplus.com*, <http://www.cplusplus.com>, 2002

## A Ritningar och bilder

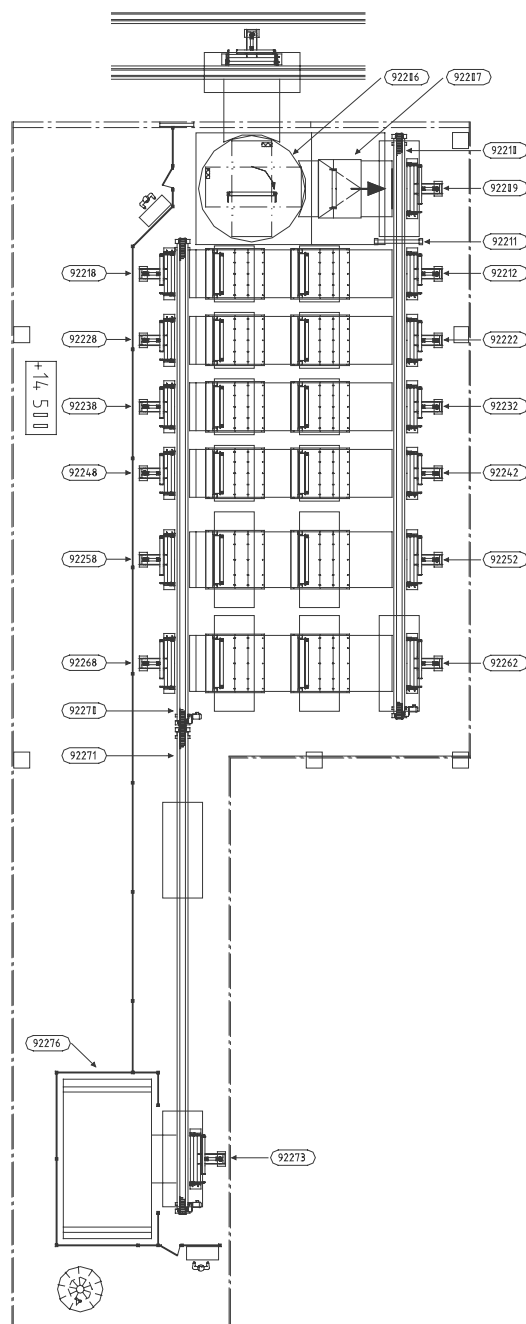
### A.1 Fabriksöversikt



*Figur 11: Översikt Braviken*

Bilden ovan visar en översikt över de olika delarna som finns i Bravikens pappersbruk. Förkortningarna PM och WH står för pappersmaskin respektive magasin (warehouse). Vi har utvecklat systemet för WH3.

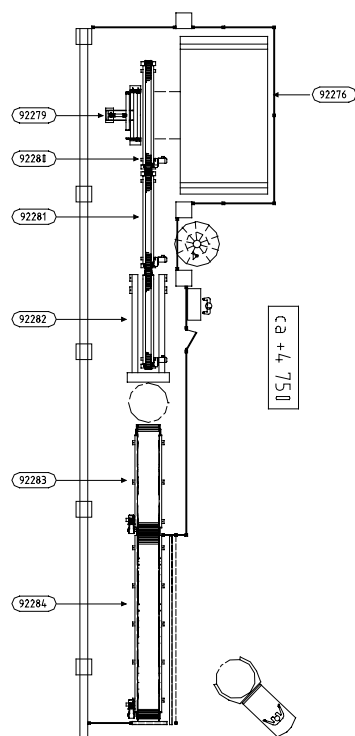
## A.2 Översikt övervåning



Figur 12: Översikt övervåning

Översikt över sorteringsbordet med maskinenheternas nummer angivna. Enhetsnumren beskrivs i bilaga B.

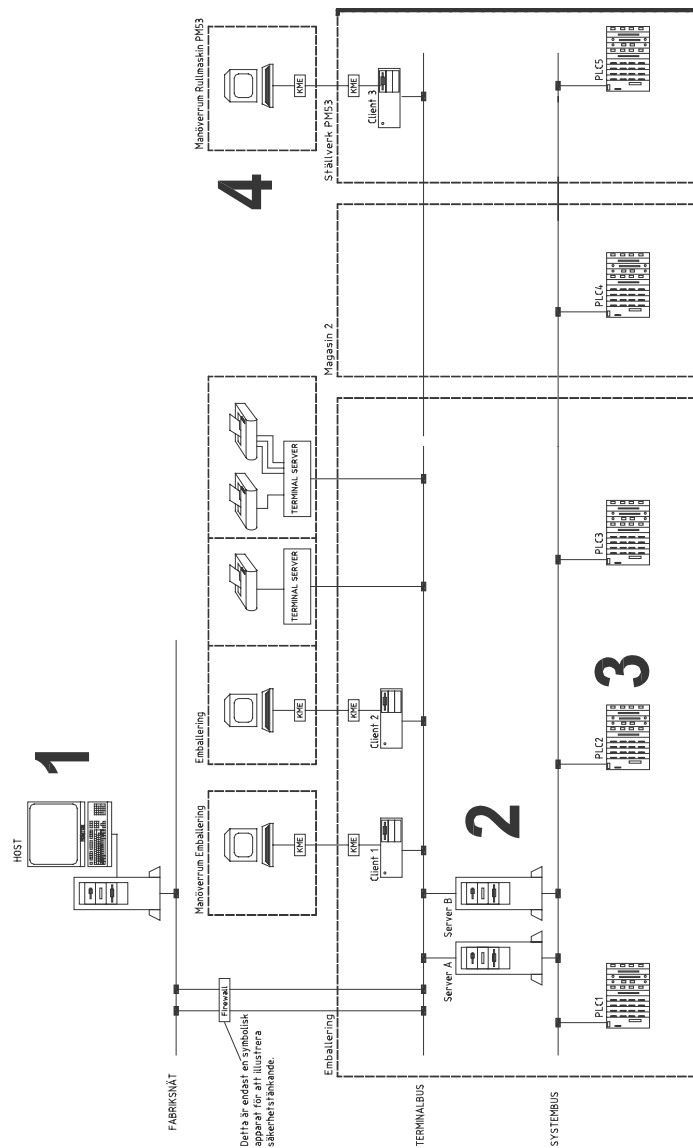
### A.3 Översikt undervåning



Figur 13: Översikt undervåning

Översikt över utmatning med maskinenheternas nummer angivna. Enhetsnumren beskrivs i bilaga B.

## A.4 Nätverksöversikt



Figur 14: Nätverksöversikt

Bilden ovan visar en översikt av nätverkskonfigurationen. De enheter som är viktiga för projekt är markerade med stora siffror och de kommer vi nu att beskriva.

1. Host är den centrala servern för det överliggande rullhanteringssystemet.
2. Server A och Server B är parallella servrar som båda innehåller serverdelen av WinCC. Server B är en identisk kopia av Server A som enbart kommer att användas om Server A kraschar.
3. PLC2 är den PLC som hanterar Magasin 3.
4. Terminal som kommer att användas för Magasin 3.

## B Enhets nummer

I denna bilaga finns alla de maskinenhets nummer som vi relaterar till i rapporten.

<b>Enhetsnummer</b>	<b>Beskrivning</b>
92206	Vridbord
92207	Lyftvagga
92209	Rullmottagare
92210	Lamelltransportör
92211	Strekkodsläsare
92212	Rullpuffare
92218	Rullmottagare
92222	Rullpuffare
92228	Rullmottagare
92232	Rullpuffare
92238	Rullmottagare
92242	Rullpuffare
92248	Rullmottagare
92252	Rullpuffare
92258	Rullmottagare
92262	Rullpuffare
92268	Rullmottagare
92271	Lamelltransportör
92272	Lamelltransportör
92273	Rullpuffare
92276	<i>Pater Noster</i> (Rullhiss)
92279	Rullmottagare
92280	Lamelltransportör
92281	Lamelltransportör
92282	Rullresare
92283	Bandtransportör
92284	Bandtransportör

*Tabell 5: Enhetsnummer*



## C Protokollspecifikationer

### C.1 Brevlåda för rullinformation

#### C.1.1 Struktur för brevlådan

Namn	Typ	Offset	Storlek	Beskrivning
BFlagsToPLC	BYTE	0	1	Används inte
bDataPointerToPLC	BYTE	1	1	Pekare
bFlagsFromPLC	BYTE	2	1	Används inte
bDataPointerFromPLC	BYTE	3	1	Pekare
SzPackId	ASCII	4	12+2	PackID
Wdestination	ASCII	18	2	'W1', 'W2', 'W3' eller 'R'
SzStorLoc	ASCII	20	6+2	Lagerplats
szOrderNo1	ASCII	28	6+2	Ordernummer
szOrderNo2	ASCII	36	6+2	Ordernummer
DwWidth	ASCII	44	4	Rullbredd [mm]
DwWeight	ASCII	48	4	Rullvikt [kg]
WrampNo	ASCII	52	2	Ramp på sorteringsbord

Tabell 6: Protokollstruktur brevlåda

#### C.1.2 Användning

*bDataPointerToPLC* fylls i av WinCC med den pappersrullpekare som information önskas om. WinCC väntar sedan till dess att *bDataPointerFromPLC* satts till samma värde som aktuell *bDataPointerToPLC*, varefter själva informationen kopieras och *bDataPointerToPLC* och *bDataPointerFromPLC* sätts till 0 av WinCC. Detta för att meddela att kopieringen är klar.

De steg som följs är:

1. WinCC skriver pekare till *bDataPointerToPLC*
2. PLCn uppdaterar informationen och skriver samma pekare i *bDataPointerFromPLC*
3. WinCC läser informationen och skriver 0 i *bDataPointerToPLC*
4. WinCC skriver 0 i *bDataPointerFromPLC* (annars fungerar inte *Upon Change*)

## D Kravspecifikation

### D.1 Inledning

#### D.1.1 Allmänt

Denna kravspecifikation ska beskriva och dokumentera de krav och mål som ska uppnås och hanteras av vårt system. Mycket tid har lagts ner på att få en så bra och djup kravspecifikation som möjligt då detta är den bästa vägen till att få ett så väl fungerande och korrekt system som möjligt.

### D.2 Översikt

#### D.2.1 Problem som systemet ska lösa

Detta systems syfte är att ge en överskådlig bild av magasinering av pappersrullar. I vårt fall inkluderar detta en vy över ett sorteringsbord samt en vy över transporten av pappersrullarna därifrån. Systemet vänder sig i första hand till operatörerna och truckförarna som befinner sig på denna avdelning av fabriken. Systemet kommer för dessa användare att visa status för varje enskild pappersrulle samt status för varje enskild maskinenhet i dessa avdelningar.

Systemet kommer att underlätta framförallt följande saker för användarna.

- **Enkel tillgänglighet av information och data**

Systemet ska på ett enkelt sätt ge användaren information om en specifik pappersrulle eller maskinenhet. Detta ska ske genom att man klickar på vald enhet. Detta fungerar dock bara på pappersrullar och inte på maskinenheter. Maskin information visas genom att vald enhet lyser med en viss färg, röd för larm, gul för handläge och grön för automatiskt läge. På pappersrullar visas även en del information utan att man behöver trycka på pappersrullen.

- **Enkel överblick**

Systemet ska ge användaren en enkel överblick över systemets status. Med detta menas att man ska veta lägen på rörliga maskinenheter och vart pappersrullar befinner sig.

- **Magasin 3 ska kunna fjärrövervakas samt kunna fjärrstyras**

Med detta menas att man ska kunna sitta på en manöverpost någon annanstans i fabriken och fortfarande se vad som händer i Magasin 3. Detta har inte varit möjligt förut då avdelningen har styrts från manöverpaneler med tryckknappar. Man vill alltså nu kunna se status på maskinenheter utan att gå fram och titta på maskinen i verkligheten. Viss övergripande styrning av sorteringsbordet ska också finnas.

- **Manuell överskridning för att snabba upp lagerhanteringen**

Med detta menas att om till exempel en truckförare håller på att köra ut en viss order på lagret och ser att det ligger en ensam rulle kvar på sorteringsbordet och väntar ska han kunna begära tömning på den rullen så att han kan köra ut den på lagret utan att behöva vänta på den. Detta innebär att man manuellt kan styra pappersrullarnas sortering.

## **D.2.2 Omgivande systems miljö**

Detta är något som vårt system kommer att få fortlöpande information om fast då indirekt genom att vår PLC pratar med det överliggande rullhanteringssystemet. Rullhanteringssystemet innehåller data över alla pappersrullar som befinner sig någonstans i fabriken.

## **D.3 Systembeskrivning**

### **D.3.1 Funktioner**

Här följer en lista och en beskrivning av de funktioner och tjänster som ska finnas i systemet. Många av beskrivningarna kommer att innehålla enhetsnummer. För placering av enheterna se bilaga A Ritningar och bilder.

#### **D.3.1.1 Prioritet 1**

- **Dynamisk storlek på pappersrullar på transportörer**

Detta måste finnas på transportör 92210. För övriga transportörer är detta önskvärt, men inte ett måste.

- **Pappersrullar som befinner sig i *Pater Noster* ska visas**

Pappersrullar som befinner sig i *Pater Noster* 92276 ska visas på samma sätt som övriga pappersrullar. Detta innebär att rullar inte på något sätt får vara dolda under tiden som de hissas ner.

- **”Tooltip” information**

För alla pappersrullar som visas på skärmen ska, när man placerar musen på pappersrullen, rampnumret visas som en så kallad ”Tooltip”.

- **Rulldata som ska visas hela tiden**

För alla pappersrullar som visas på skärmen ska ordernummer och destination i lagerbyggnaden visas. Det ordernummer som visas behöver inte vara det fullständiga, utan det räcker med de 6 sista tecknen i ordernumret.

- **Det ska gå att klicka på synliga rullar för ytterligare data**

Alla pappersrullar som visas på skärmen ska man kunna klicka på med musen. Då man klickar på en pappersrulle ska ytterligare information visas.

Den önskade informationen är:

- PackID – fullständigt rullidentifikation (12 tecken).
- Destination – vilket magasin rullen kommer att lagras i.
- Ordernummer – fullständigt ordernummer (12 tecken).
- Lagerplats – placering i lager.
- Rampnummer – vilket sorteringsbord och vilken ramp som denna pappersrulle passerat.
- Rullvikt – pappersrullens vikt i kg.
- Rullbredd – pappersrullens bredd i mm.

- **Val av språk**

Systemet ska gå att köra med svenska och engelska texter. Omkopplingen mellan dessa ska ske via knappar som alltid är synliga.

- **Säkerhetsområden ska visas**

Avgränsning och status för de säkerhetsområden som finns i Magasin 3 ska framgå av bilderna. Vid utlöst säkerhetsområde ska motsvarande avgränsning färgas röd. Vidare ska det gå att via knappar sätta varje säkerhetsområde i manuellt läge. Eftersom systemet är uppdelat på två vyer ska det finnas en lampa som indikerar om något säkerhetsområde har utlöst. Detta för att veta om ett område på den andra bilden är utlöst.

- **Körlägesområden ska visas**

Avgränsning och status för de körlägesområden som finns i Magasin 3 ska framgå av bilderna. Vidare ska det gå att via knappar sätta varje körlägesområde i manuellt eller automatiskt läge.

- **Styrning av hydraulik**

Det ska gå att övervaka de hydrauliska enheter som finns i Magasin 3. Viss styrning ska kunna ske. Den styrning som ska finnas är av/på av hela hydrauliken samt av/på för varje separat pump som finns i systemet.

- **Visa status om varje ramp på sorteringsbordet**

För varje ramp, 1 till 6 på sorteringsbordet ska information visas om tömning pågår, om tömning är begärd, vilket ordernummer och vilken lagerplats som körs på rampen, tid kvar till timeout samt om rampen är till- eller frånslagen.

- **Larmvisning**

Det ska visas på alla bilder om det går ett larm någonstans. Larmet ska ge information om följande:

- Enhetsnummer
- Vilket område, alltid WH3 för oss
- Enhetens namn i text, ex Transportör
- Enhetstyp (BMK), ex M = motor, B = givare, Y = ventil
- Beskrivning av felet, ex om det blivit en timeout
- Vilken PLC som har satt felet
- Flagadress, larntaggens adress i PLCn

#### D.3.1.2 Prioritet 2

- **Taggar ska vara i strukturer**

För att underlätta integrationsarbetet ska Taggar placeras i strukturtaggar. Bara i undantagsfall får ensamma Taggar förekomma.

- **Visning av enheter vid övergång till manuellt säkerhetsområde**

En lista över de enheter som ännu inte har hunnit byta läge från automatiskt till manuellt läge ska visas.

#### D.3.1.3 Prioritet 3

- **Rörliga enheters aktuella läge ska visas**

I bilderna ska det framgå i vilket läge de rörliga enheterna befinner sig. De rörliga enheter som finns i Magasin 3 är vridbord 92206 och uppresare 92282.

- **Styrning av ramperna 1-6**

Man ska kunna begära tömning av en eller av samtliga ramper.

#### D.3.1.4 Prioritet 4

- **Nödstopp**

Om ett nödstopp är utlöst ska det framgå vart ifrån det kommer ifrån.

- **Larmnummer**

Applikationen ska använda sig av larmnummer mellan 200 000 – 299 999.

#### D.3.1.5 Prioritet 5

- **Operatörsmanual**

En operatörsmanual ska framställas. Manualen ska behandla systemets uppbyggnad och handhavande så som det är relevant för en användare. Den ska skrivas på svenska.

#### D.3.2 Prestandakrav

- **Uppdateringshastighet för dynamiska pappersrullar**

Under drift kommer pappersrullar på transportörer att flyttas 0,25 m per sekund. På bilderna får inte pappersrullarna visas med en förskjutning överstigande 0,5 m mot verkligheten.

- **Dokumentera och minimera nätverkstrafik**

Den nätverkstrafik som uppkommer vid kommunikation med PLC ska minimeras. Vidare ska den resulterande trafikmängden uppskattas och dokumenteras. Den trafikmängd som kan tillåtas är inte riktigt bestämd men ska helst inte vara mer än 10kbyte/s. Detta beroende på att hela fabriken använder samma nätverk vilket ger en belastning. Det är alltså viktigt att försöka minimera belastningen.

#### D.3.3 Övrigt

- **Användarvänligt**

Systemet ska vara utformat så att det upplevs som användarvänligt och enkelt att komma igång med. Det ska inte krävas någon större arbetsinsats att lära sig de grundläggande funktionerna.

- **Skärmstorlek**

De grafiska terminaler som används kommer att ha 19” skärmar med skärmapplösningen 1280x1024. Med dessa förutsättningar ska samtliga texter, ikoner och dylikt på skärmbilderna ska vara lätt synliga och läsbara.

## E Lamb AGs riktlinjer för WinCC projekt

### E.1 Uppbyggnad

Skärmen delas upp i tre avsnitt. Dessa är:

- Larm

Ligger längst upp på skärmen och innehåller larmlistan.

- Process

Är placerad mellan larm och meny. Innehåller de bilder som visar processen.

- Meny

Ligger längst ner på skärmen och innehåller menyval, exempelvis språkbytesknappar.

### E.2 Lager

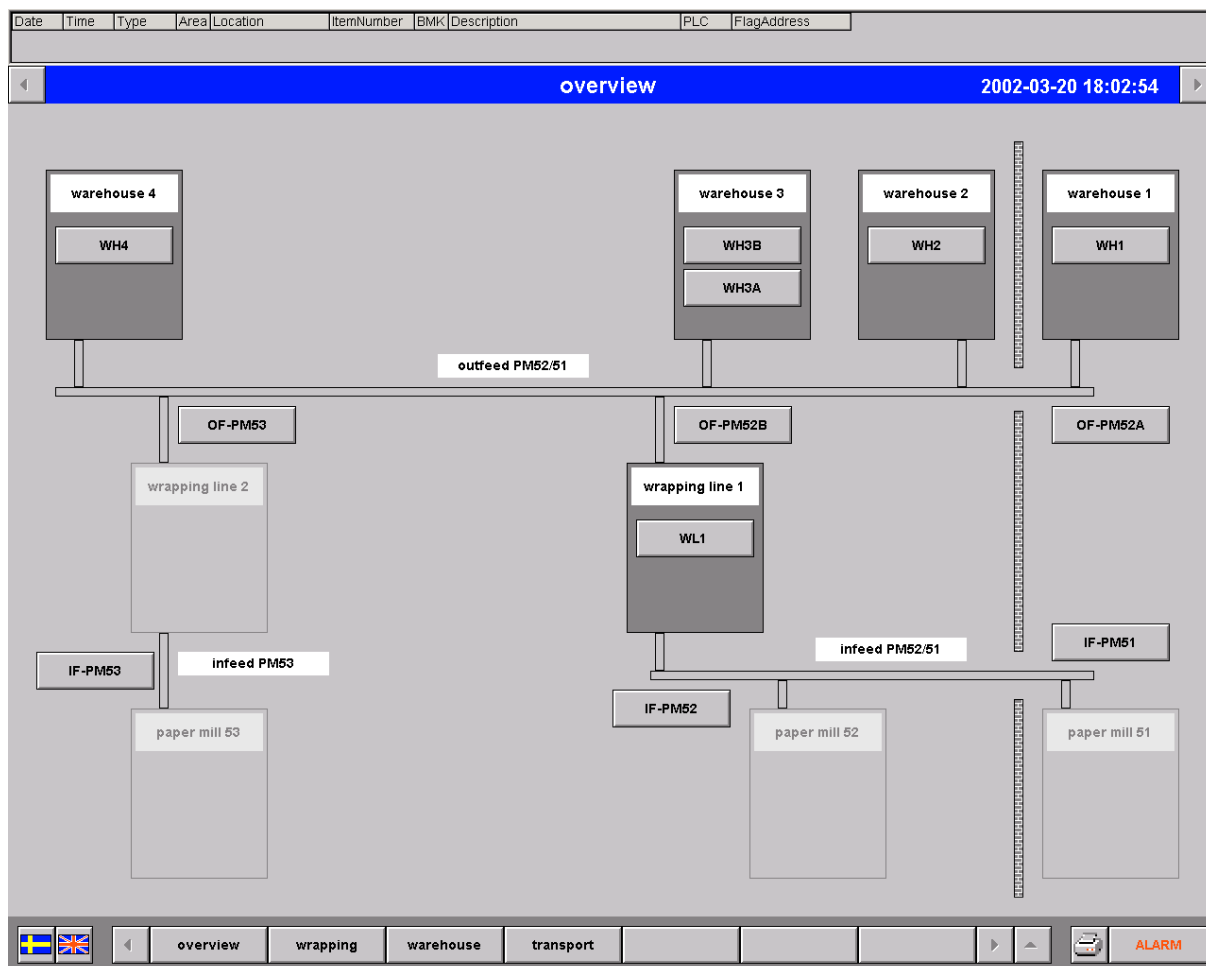
I WinCC har man 16 olika bildlager att jobba med, där lager 0 är längst ner och lager 15 högst upp. Dessa ska användas enligt följande tabell.

Lager	Beskrivning
0	Platshållare för fönster.
1-5	Statiska bakgrundsbilder.
6-7	Statiska textobjekt länkade till interna Taggar (för att kunna byta språk).
8	I/O-texter, Statusindikeringar
9	Vanliga dynamiska pappersrullar
10	Mynt för dynamiska pappersrullar
11	Säkerhetsområden
12	Objekt med dynamiska attribut
13	Reserverad
14	Knappar
15	Reserverad

Tabell 7: Lamb riktlinjer

## F Systembilder under körning

### F.1 Översikt, startbild



Figur 15: Översikt, startbild under körning

Denna bild är den som binder samman alla delarna av rulltransportsystemet. Den är också startbild för de grafiska terminalerna och det är från alla andra bilder enkelt att ta fram denna. För att studera ett specifikt område klickar man på den del man är intresserad av.

Den del av systemet som vi utvecklat nås genom att klicka på någon av knapparna i ”warehouse 3”, magasin 3. Knappen märkt WH3A leder till övervåningen och knappen märkt WH3B leder till undervåningen.

Bilden är tagen då systemet är i engelskt läge.



## F.2 Sorteringsbordet

>>Date >>Time >>Type >>Ar >>Location >>ItemNumber >>BN >>Description >>PLC >>FlagAddress

magasin 3A 2002-03-19 14:26:51

tillfrån	ramp 1	ramp 2	ramp 3	ramp 4	ramp 5	ramp 6
	till	till	till	från	till	från
orderNr	111111	222222	333333		444444	
rack	AAAAA	BBBBB	CCCCC		DDDDD	
timeout	20	0	10	0	5	0
	töm	töm	töm	töm	töm	töm
		tömning pågår			tömning begärd	

driftläge

säkerhetsomr.

hydraulik

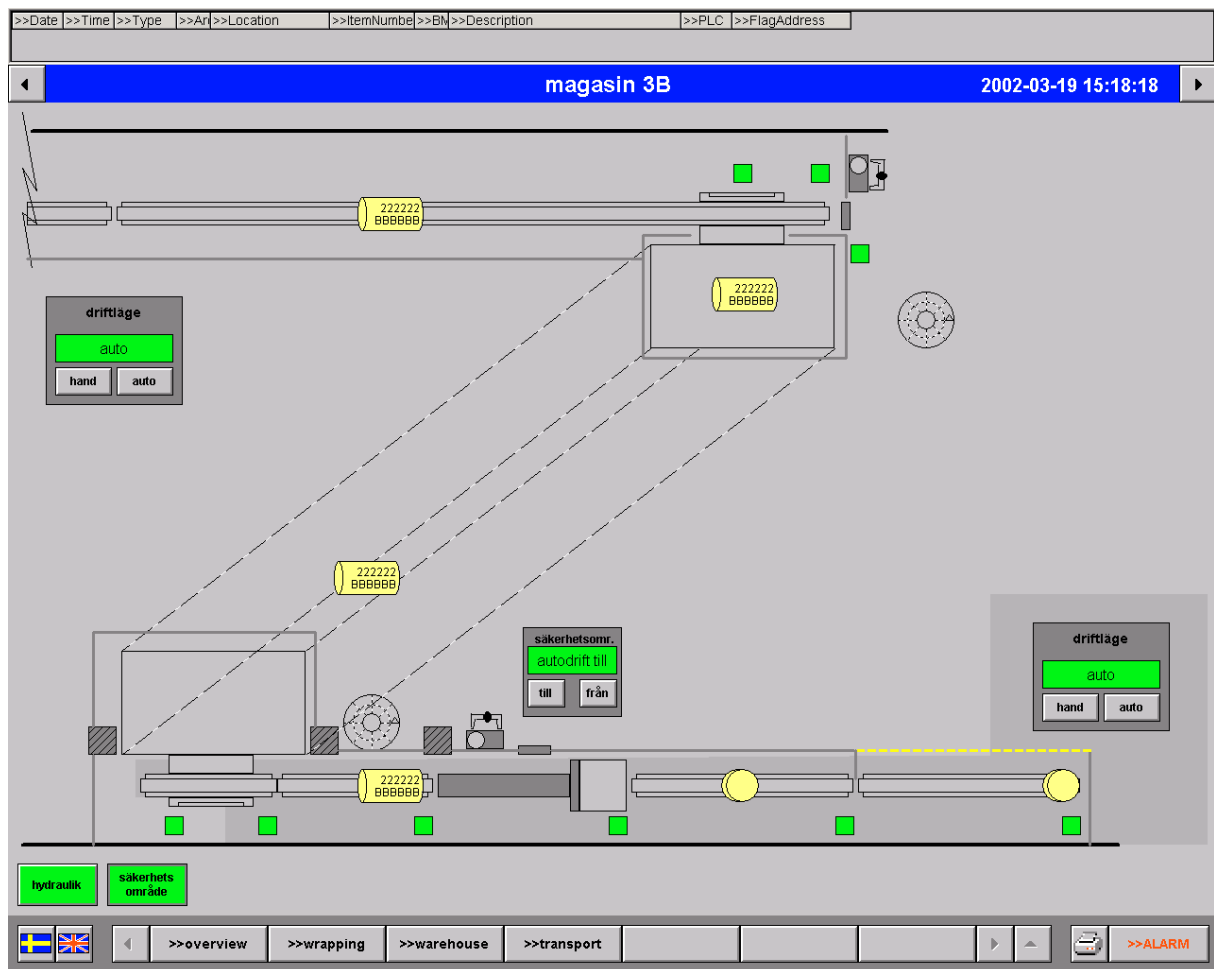
tömning av alla ramper

>>overview >>wrapping >>warehouse >>transport >>ALARM

Figur 16: Sorteringsbord under körning

Denna bild visar en typisk situationsbild av sorteringsbordet. En pappersrulle är på väg in (pappersrullen längst till vänster) och två pappersrullar är på väg ut mot undervåningen (de som är på den undre transportören).

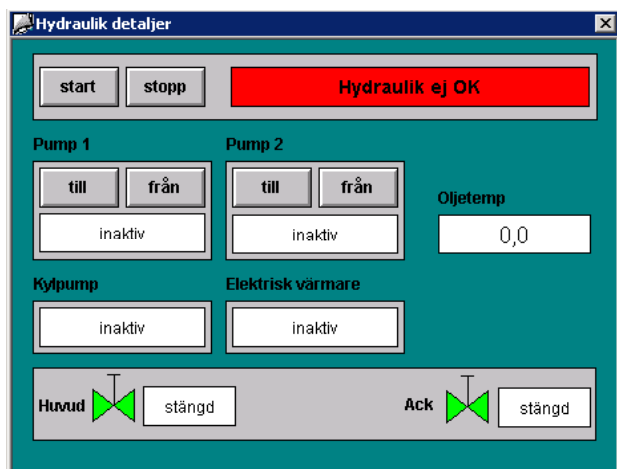
### F.3 Utmatning



Figur 17: Utmatning under körning

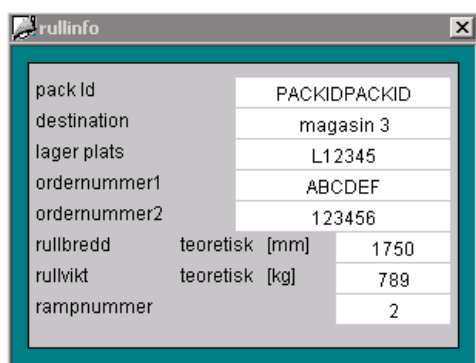
Denna bild visar undervåningen under drift. En pappersrulle kommer precis från sorteringsbordet, två befinner sig under nedhissning i *Pater Nostern* och en är på väg in i rullresaren. Två par redan uppresta pappersrullar kan snart hämtas av truck för transport till rätt plats i själva magasinet.

## F.4 Hydraulik



Figur 18: Hydraulik detaljer under körning

## F.5 Informations visning av pappersrulle



Figur 19: Information om pappersrulle under körning

## G Begrepp

### Action

En händelse som inträffar eller utförs i systemet. Till exempel att någon klickar med musen på ett objekt.

### ActiveX

Språkoberoende teknik i Microsoft Windows för mjukvarukomponenter.

### C-script

Det skriptspråk som används i WinCC. Bygger på C.

### HMI

*Human Machine Interface* Uttryck för vad gränssnittet mellan maskiner och användare kallas.

### HTML

*Hyper Text Markup Language*. Standard för hur man skapar webbsidor.

### Ladderdiagram

Ett speciellt sätt att programmera en PLC på då man använder kretsliknande bilder för att skapa programmet.

### MPI

*Multi Point Interface*. Seriell kommunikationsbuss, typisk hastighet 1,5-15Mbps.

### ODBC

*Data Sources Open Database Connectivity*. Databasabstraktionslager.

### OLE

*Object Linking and Embedding*. En standard i Microsoft Windows för att dela objekt mellan applikationer.

### Pater Noster

*Lowerator*. En maskinenhet för att hissa ner last. Består av lastfickor som går runt på ett par kedjor. Lastfickorna fylls på övre delen och töms på undre delen av enheten.

### PLC

*Programmable Logic Controller*. Styr dator för fabrik.

## PLC-cykel

Den tid ett varv i PLCns huvudprogramslina tar. Detta rör sig i storleksordningen millisekunder. Huvudslinan består av tre olika steg. Det första som händer är att alla ingångar läses av. Efter detta exekveras huvudprogrammet som hanterar datan och slutligen skrivs information till utgångarna.

## PROFIBUS

Öppen fältbuss standard

## Simatic S7

Siemens PLC familj.

## Simatic STEP7

Programutvecklingsmiljö för Simatic S7.

## SQL

*Structured Query Language*. Tillverkarneutralt frågespråk för relationsdatabaser.

## Struktur tag

Gruppering av enskilda Taggar. Jämför structar i C/C++.

## Tag

Global variabel i WinCC som antingen är kopplat till ett dataområde i en PLC eller fungerar som en intern variabel.

## Tooltip

Liten textruta med information som visas när muspekaren förs över ett objekt.

## TCP/IP

*Transmission Control Protocol/Internet Protocol*. Protokoll, som framförallt används på Internet.

## Variabel

Ett namngivet lagringsutrymme i datorns minne.

## WinCC

WinCC är en industri- och teknologineutral utvecklingsmiljö med inriktning mot visualiserings- och processkontrolleringsapplikationer inom produktions- och processautomationsmiljöer.

## **H Användarmanual**

# Magasin 3

<b>PRIORITERING VID SORTERING OCH TÖMNING .....</b>	<b>3</b>
PRIORITERINGSORDNING VID SORTERING.....	3
PRIORITERINGSORDNING VID TÖMNING.....	3
<b>GENERELLA EGENSKAPER HOS BILDERNA.....</b>	<b>4</b>
<b>BILD 1 AV 2 - SORTERINGSBORD .....</b>	<b>6</b>
<b>BILD 2 AV 2 - UTMATNING.....</b>	<b>8</b>



## **Prioritering vid sortering och tömning**

### ***Prioriteringsordning vid sortering.***

I nedanstående ordning söks möjliga fack upp för nya rullar:

Hänsyn tas hela tiden till mått, ordernummer och lager placering.  
Rullar större än 2100 mm går endast till fack 5 och 6.

1. Fack som redan har rullar med samma ordernummer och plats för fler rullar.
2. Tomma fack eller fack som inte är tilldelade något ordernummer.

När blandning är nödvändig tas hänsyn till befintliga rullar i facket. Om befintlig rulle i facket är mindre än 1200 mm får nästa rulle inte vara bredare än 1,5 x den befintliga rullens bredd.

3. Fack med färdig trucklast och ej blandade.
4. Fack med utlöpt väntetid och ej blandade.
5. Alla fack som inte är blandade.
6. Alla fack som har plats.

Efter att den nya packlinjen har tagits i drift, kommer också programmet att ta hänsyn till rullkön från packen. Blandning görs då i första hand på ordrar/fack som inte kommer att fyllas på med fler rullar.

### ***Prioriteringsordning vid tömning.***

I nedanstående ordning söks möjliga fack upp för tömning.

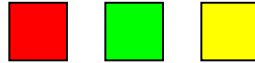
1. Manuellt inställt val av fack.
2. Fack som har full trucklast och 8 eller fler rullar.
3. Fack som har 8 eller fler rullar.
4. Komplet trucklast på samma fack som senast tömdes.
5. Exakt en trucklast i fack och ingen rulle på väg på T3.
6. Fack med full trucklast och blandade order.
7. Fack med full trucklast
8. Fack som har blandade ordrar.
9. Fack med utlöpt väntetid.

## Generella egenskaper hos bilderna

### 1. Maskinstatus

Alla maskinenheter visar sin status med hjälp av kvadratiska lampor som sitter bredvid varje enhet. Dessa lampor har följande indikeringsfärg

- Röd – Larm
- Grön – Autoläge
- Gul – Handläge

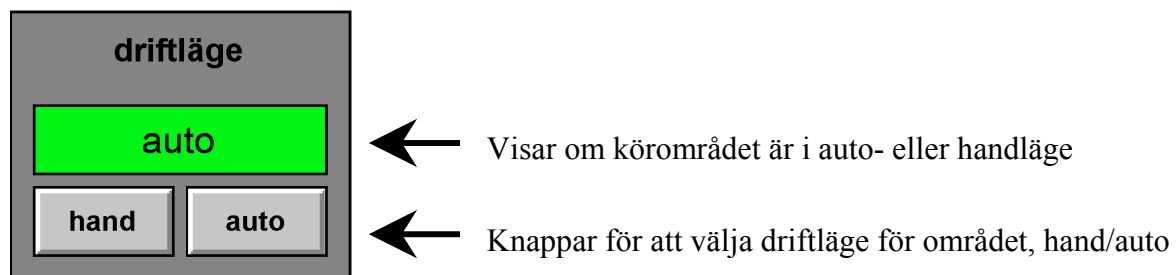


### 2. Visning av enhetsnummer

För att visa enhetsnumret för en maskinenhet för man musen över aktuell maskins status lampa. Då detta görs kommer en liten ruta dyka upp som visar aktuellt maskinnummer.

### 3. Driftläge

Varje körområde som finns, dvs. 1 (sorteringsbordet), 2 (utmatning och paternoster på plan 14,5 samt rullmottagare 92279 på plan 4,75) och 3 (plan 4,75 utom just rullmottagare 92279), har en varsin panel där status för området kommer att visas. I panelen kan man även sätta området i Autoläge eller i Handläge.



### 4. Larmområde

- Larmområdena på de olika bilderna kommer att vara markerade mörkgrått då inget område har blivit aktiverat. Då ett område blir aktiverat genom att någon bryter en ljusbom, öppnar en grind eller trycker på ett nödstopp så kommer detta område att lysa rött.
- Om en grind är öppen kommer denna att blinka mellan grönt och gult annars är den markerad i grått.
- Om ett nödstopp aktiveras kommer detta att blinka mellan grönt och gult.
- På varje bild finns även en lampa längst ner till vänster, märkt med ”säkerhetsområde”. Denna lampa lyser rött om något säkerhetsområde i magasin 3 har blivit utlöst. Om inget säkerhetsområde har utlösts lyser lampan grönt.
- Varje säkerhetsområde har även en panel där man kan se statusen på området samt sätta det av eller på. Då man slår av området kommer en ruta att dyka upp där man ser en lista över de maskinenheter som ännu inte gått över till manuellt säkerhetsläge. Dessa panel ser ut så här:



← Visar om säkerhetsområdet är till eller från  
← Knappar för till- respektive frånslagning av säkerhetsområdet

## 5. Hydraulik

Längst ner till vänster på varje bild finns en knapp märkt "hydraulik". Denna knapp är färgad grön om hydrauliken är ok och röd om ett larm har utlöst.

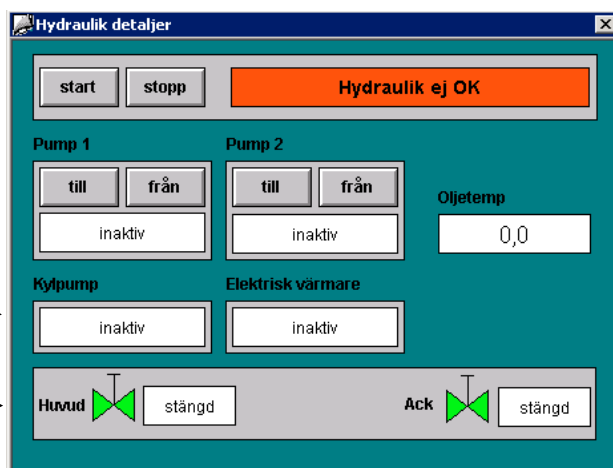
Då man klickar på denna knapp visas en ruta med styrning och information om hydrauliken. Denna ruta ser ut så här:

Knappar för start och stopp av hydrauliken

Till och frånknappar samt status för pump 1 och 2s driftlägen

Visar driftläge för kylpump, oljevärmare

Visar huvudventilens läge



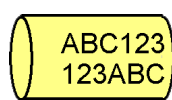
← Visar status för hydraulikenheten

← Visar den aktuella oljetemperaturen

← Visar ackumulator ventilens läge

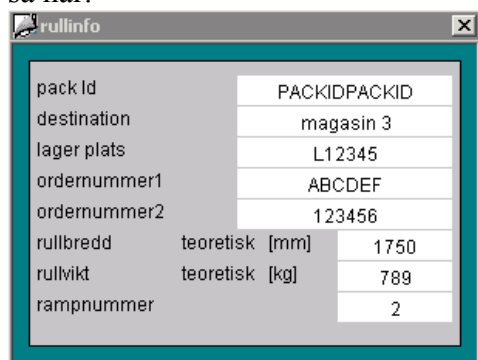
## 6. Allmänt för pappersrullar

Alla pappersrullar som visas på skärmen visar vilket ordernummer samt vilken lagerplats pappersrullen ska till.



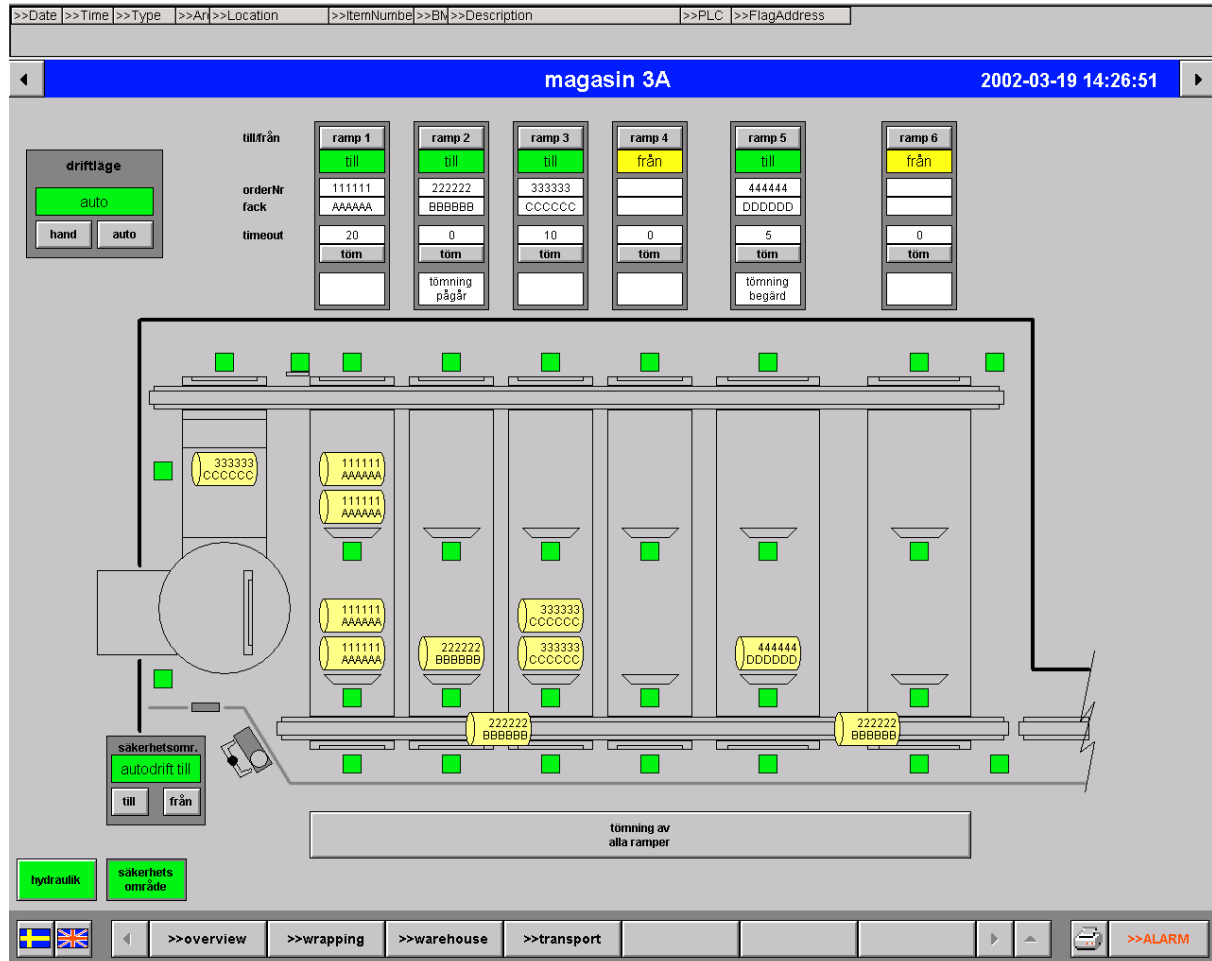
← Visar ordernummer  
← Visar lagerplats

För att få fram fullständig information för en pappersrulle klickar man på den med musen. Då man gjort detta öppnas ett nytt fönster med den kompletta informationen. Detta fönster ser ut så här:



## Bild 1 av 2 - Sorteringsbord

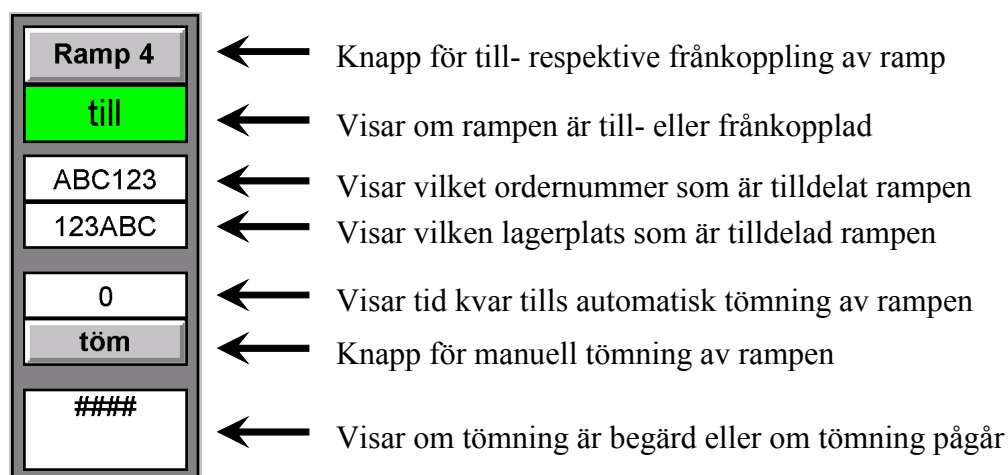
Den här bilden visar första delen av plan 14,5 i magasin 3.



På denna bild visas inmatning, sorteringsbord och början av utmatningen.

Driftlägespanelen och säkerhetsområdespanelen på vänstra sidan samt lamporna längst ner till vänster beskrivs i den generella delen.

Överst i bilden ligger panelerna för styrning och visning av information för ramperna. Varje panel är uppbyggd på följande sätt:



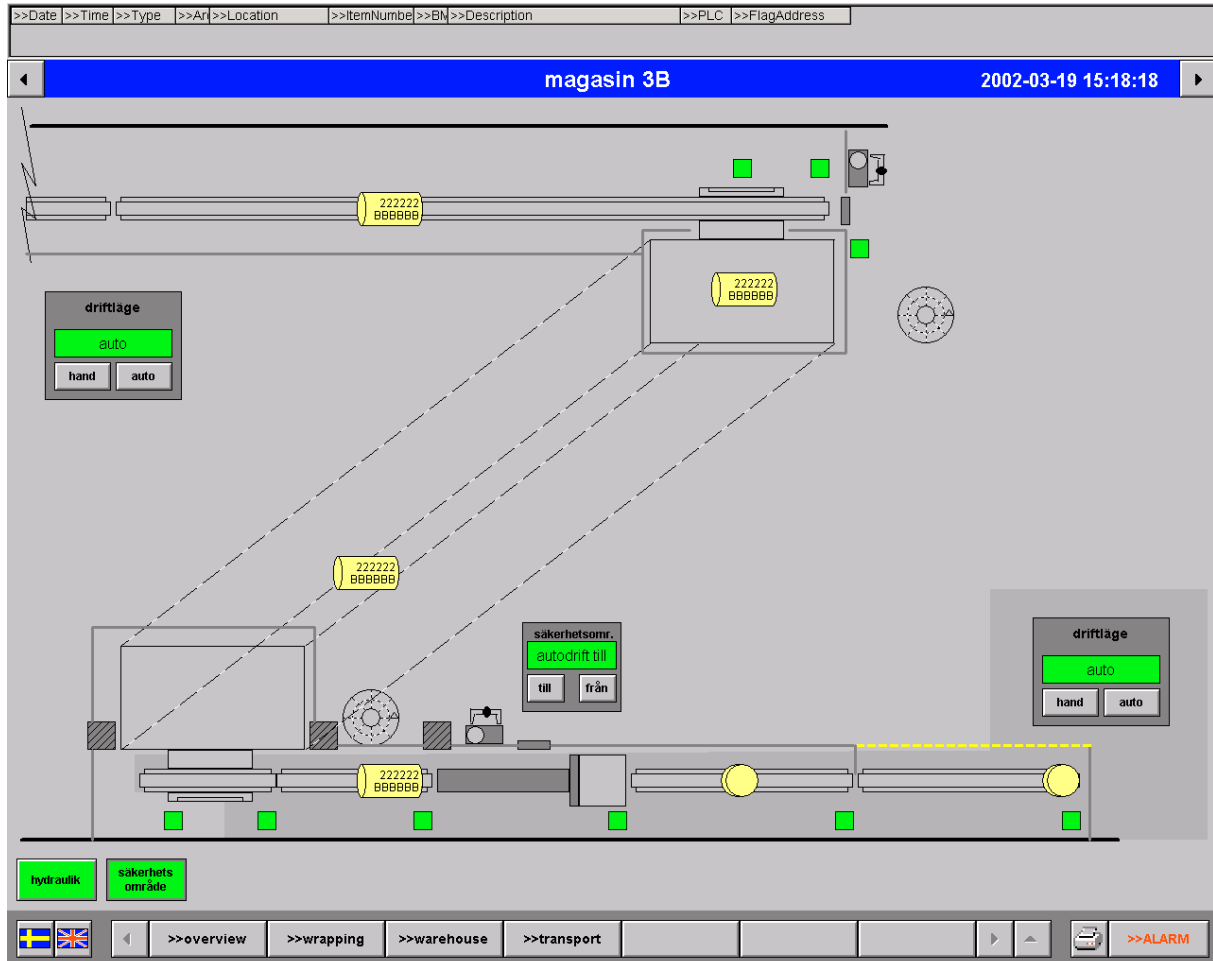
Då man begärt manuell tömning av en ramp genom att klicka på ”töm”-knappen, visas en varningsruta där man måste bekräfta eller avbryta begäran.

I mitten av bilden visas hur rullarna ligger på sorteringsbordet och under sorteringsbordet ligger utmatningstransportören.

Längst ner på bilden finns en knapp för att begära tömning av alla ramper. Om man klickar på denna visas en varningsruta där man måste bekräfta eller avbryta begäran.

## Bild 2 av 2 - Utmatning

Denna bild visar resterande del av övervåningen, pater noster samt undervåningen.



På denna bild visas en transportör på plan +14,5, pater noster samt alla enheter på plan +4.75. Driftlägespanelen och säkerhetsområdespanelen på vänstra sidan samt lamporna längst ner till vänster beskrivs i den generella delen.

De rullar som befinner sig under transport i pater noster visas på diagonalen i bildens mitt. Efter rullresaren ses rullstaplarna uppifrån och man kan se antalet cirklar om stapeln innehåller en eller två rullar.

På transportör 92281 indikeras full transportör genom att en rullstapel visas i transportörens högra ände. Den gula linjen som syns på bilden indikerar att en truck håller på att hämta rullar.