Computer Science

Patrik Andersson och Björn N. Lindhe

Grafisk presentation av övervakade nätverk

Bachelor's Project

2004:05

Grafisk presentation av övervakade nätverk

Patrik Andersson och Björn N. Lindhe

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 vårt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Patrik Andersson

Björn N. Lindhe

Godkänd, 2003-01-15

Handledare: Mari Göransson

Examinator: Stefan Lindskog

Sammanfattning

Idag när företag växer, men samtidigt behöver dra ner på personal är de i stort behov av automatiserade processer inom företaget. Samtidigt behöver företag ha möjlighet till snabba åtgärder ifall olyckan skulle vara framme. Det här projektet inriktar sig på den IT-relaterade delen av ett företags organisation med fokus på övervakning av nätverksutrustning. Projektet som sådant är uppdelat i två mindre projekt då en fullständig implementation av hela programvaran skulle ta alldeles för lång tid. De två delarna är dels den delen där insamlingen av datan från nätverket sker med lagring av datan i databas och dels ett där presentationen av den insamlade datan sker - vilket behandlas här.

Rapporten tar upp och beskriver de olika delar i övervakningen som möjliggör presentation av den insamlade datan. Från design och implementation av den databas, där all insamlad data lagras, till de olika tekniker och möjligheter som finns för design och implementation av användargränssnittet. Den här delen av huvudprojektet har resulterat i en applikation med ett användargränssnitt innehållande både en översiktlig och en mer detaljerad nivå med information om de olika övervakade noderna i ett nätverk.

Graphic Presentation of Monitored Networks

Abstract

Today when companies grow larger, but at the same time need to cut their number of employees they are in great need of automated processes inside the company, but with the possibility of quick remedies if an accident should occur. This project concentrates on the IT-related part of a company's oganization with focus on the surveilance of networks.

A complete implementation of the whole application would take far more time than available. The project has therefore been divided into two smaller projects - one where the collection of data from the network is processed, including the insertion of data into the database, and one where the presentation of the collected data is done, which is presented here.

The project describes the different parts in the surveilance that makes the presentation of the collected data possible; from design and implementation of the database, where all collected data is stored, to the different techniques available for design och implementaion of the user interface. In the end of the report there is a summary and a presentation of the conclusions made during the project. The project has resulted in an application with a user interface for both a general and a more detailed level of information about the surveyed nodes in networks.

Innehåll

Figurer

Tabeller

1 Inledning

Arbete med övervakning av nätverk kräver noggrann planering om vilka data som skall samlas in. Därför sammanställdes i början av projektet ett dokument där specifikation om vad för data som skulle samlas in presenterades (se appendix F). Kapitel två börjar med att beskriva huvudprojektet och indelningen i de två delprojekten. Kapitlet tar även upp vilka krav som ställs på den slutliga produkten för det här delprojektet och vad målet med det är.

Påföljande kapitel ger en kort introduktion till SNMP, som används vid insamling av data i nätverk. Kapitlet beskriver även i korthet olika tekniker för insamling av data och även hur "icke begärd" data kan skickas från klient till server.

Kapitel fyra beskriver design och implementation av den databas som används för att lagra insamlad data. I kapitlet ges en kort introduktion till relationsdatabaser med beskrivning om vilka olika typer av relationer som kan existera i dem. I den senare delen av kapitlet beskrivs den slutgiltiga designlösningen, samt några av de alternativa lösningar som diskuterades under projektets gång. Slutligen presenteras en kort beskrivning av vilka tekniker som användes vid implementationen.

Kapitel fem presenterar design och implementation av det grafiska användargränssnittet. Kapitlet tar även upp olika tekniker för design och skärmdisposition, samt olika frågeställningar som bör övervägas innan själva designen påbörjas. I det sista kapitlet presenteras en sammanfattning av projektet och de slutsatser det lett fram till.

2 Beskrivning av projektet

Idén till projektet grundas i att det idag bara finns ett fåtal bra programvaror som klarar av att övervaka ett nätverk, samt presentera den insamlade informationen på ett snyggt och lättläst sätt. Programvarorna är oftast dyra och besvärliga att administrera för den vanliga användaren. Det här projektet har därför inriktats på att skapa samma typ av programvara, men samtidigt göra den lättförståelig och lättadministrerad.

I det här kapitlet presenteras en övergripande beskrivning av projektet, samt de mål, avgränsningar och krav som specificerats. I kommande kapitel presenteras en mer djupgående förklaring om de olika delarna av projektet och hur tankegångarna har varit kring de lösningar som slutligen använts vid implementationen.

2.1 Huvudprojektet

Huvudprojektet är ett projekt som grundats av Patrik Andersson på företaget Ataco i Karlstad. Det går ut på att skapa ett komplett övervakningssystem för nätverksmiljöer övervakade med SNMP (Simple Network Management Protocol). Idén till huvudprojektet grundades i att det idag är ont om enkla, bra och billiga system för övervakning av utrustning såsom servrar, arbetsstationer, switchar med mera i nätverk. Många av de system som existerar är oftast väldigt dyra och innehåller allt för många komplexa handgrepp innan önskad information kan presenteras. I det här projektet har därför fokus lagts på att göra informationen så lättåtkomlig, lättöverskådlig och lättläst som möjligt.

I och med projektets storlek har det delats in i två mindre delprojekt. Det första delprojektet innefattar den del i huvudprojektet som har med själva insamlingen av datan att göra. Projektet behandlar därmed all inhämtning av SNMP-data från de övervakade noderna till den övervakande centralnoden, samt lagringen av datan på rätt plats i databasen.

Det andra delprojektet, som behandlas i det här arbetet, innefattar presentation av den lagrade informationen. En närmare beskrivning presenteras i nästa delkapitel.

2.2 Delprojektet

Delprojektet går ut på att skapa ett grafiskt användargränssnitt (GUI)[1] för presentation av den information finns som lagrad i databasen. Utöver användargränssnittet ingår även att designa och implementera den databas som datan från de övervakade noderna lagras i. Anledningen till att det här delprojektet innefattar desigen av databasen är att arbetet mellan de två delprojekten blir mer jämt fördelat, samt att designen på databasen starkt påverkar vilken information som är möjlig att presentera.

2.3 Syfte och mål

Målet med huvudprojektet är att skapa en serverövervakningsmiljö baserad på SNMP, PHP och mySQL. Målet med delprojektet är att designa och implementera en databas för lagring av insamlad data. Utöver databasen skall ett grafiskt användargränssnitt designas och implementeras där information om de övervakade noderna presenteras på ett strukturerat och lättöverskådligt sätt. Syftet är även att undersöka och dokumentera de frågeställningar som framkommer i rapporten. Detta för att ge gemene man en grundläggande insyn i systemets funktion och uppbyggnad. Det är dock inte tänkt som en manual för systemet.

2.4 Avgränsning

För att närmare klargöra vad som inte kommer att ingå i det här delprojektet kommer här en lista baserad på de punkter vi inte har för avsikt att ta upp:

- Vilka metoder som kommer att användas för insamlig av data
- Lagring av insamlad data i databasen
- Hur meddelanden skickas mellan noder i nätverket
- Hur SNMP-paket är uppbyggda

[1] Graphical User Interface

- Hur data i SNMP-paketen kommer att tas om hand

Ovan nämnda punkter förväntas istället behandlas i det första delprojektet.

2.5 Kravlista

Följande lista tar upp de tekniska krav och utformningar som kommer att ställas på den färdiga produkten. Listan avser de krav och användarfall som anses vara de mest kritiska och primära.

Kravspecifikation databas:

- Databasen skall kunna lagra data som utifrån dokumentet "Förslag till insamling av SNMP-data" (se appendix F) diskuterats och enats om mellan grupperna.
- Databasen måste byggas med avseende på framtiden och skall därmed designas på ett såpass flexibelt sätt att det med rimlig arbetsinsats är möjligt att lägga till nya typer av utrustning och data.

Kravspecifikation gränssnitt:

- Gränssnittet skall vara skrivet på engelska.
- Gränssnittet skall vara anpassat för upplösningen 1280x1024.
- För att göra systemet lättöverskådligt skall övergripande information om alla typer av utrustning i nätverket presenteras på första sidan.
- För att hålla användaren ajour, skall gänssnittet med jämna tidsintervall ladda in den senaste informationen från databasen.
- Då en kritisk nivå hos en entitet har uppnåtts skall den del av gränssnittet som representerar entiteten markeras med eller byta färg till röd (alternativt annan lämplig färg).
- Då en användare väljer att visa detaljerad information om en entitet skall ett nytt fönster öppnas där en presentation om den aktuella entiteten, samt dess status kan överskådas.
- En historik för varje entitet i nätverket skall kunna presenteras.

2.6 Metod

Det här projektet är förhållandevis praktiskt och saknar därför en del av de teoretiska metoder som återfinns i mer undersökande arbeten. Den metod som har använts för att besluta vilka olika data som skall samlas in om de olika typer av utrustning som skall övervaka baseras på RFCer[2] och den information som kunde återfinnas i ASN.1[3], samt den MIB[4] som finns att tillgå om SNMP. Grupperna diskuterade sedan gemensamt och kom fram till vilka data som skulle samlas in. När alla källor systematiskt gåtts igenom sammanställdes dessa till en lista med den data som ansågs relevant - "Förslag till insamling av SNMP-data" (appendix F).

Metoden som användes vid skapandet av det grafiska användargränssnittet baseras mycket på "trial-and-error"-metoden. Den har i sin tur använts i kombination med sunt förnuft och regler för hur ett gränssnitt bör designas. Undersökningar har även gjorts på andra applikationer där inspiration om hur design av ett användargränssnitt kan gå till hämtats, samt hur de absolut inte bör designas.

[2]Request For Comments
[3]Abstract Syntax Notation one
[4]Management Information Base

3 SNMP

SNMP står för "Simple Network Management Protocol" och är ett klient-server-protokoll. Ordet "Simple" betyder i det här sammanhanget inte "enkel", utan innebär istället att protokollet förenklar en uppgift som annars skulle vara betydligt mer komplex [3]. Protokollet är mycket innehållsrikt och datan som går att samla in sträcker sig över ett stort område. I skrivandets stund är den senaste versionen av SNMP version 3, där bland annat kryptering av datan som skickas mellan noderna har implementerats.

I det här kapitlet presenteras grundläggande kunskaper om SNMP tillsammans med några av de metoder för hämtning av data. En kort övergripande beskrivning av protokollet ges, samt att några för och nackdelar diskuteras. Datan som samlas in via SNMP lagras i den databas som i detalj beskrivs i kapitel 4.

3.1 Beskrivning av protokollet

SNMP är ett protokoll för övervakning och konfiguration av nätverksutrustning. På varje nod i nätverket installeras en SNMP-agent, som på begäran (eller genom så kallade "traps"[5]) skickar data till den centrala insamlingsnoden. Övervakning och konfiguration utförs med hjälp av fem olika kommandon [2];

get hämtar det efterfrågade värdet. Att hämta ett värde på det här sättet är väldigt effektivt, om man jämför med alternativ som exempelvis Telnet och SSH, då ingen inloggning sker för att hämta datan vid användning av SNMP.

get-next är en uppföljare på kommandot get och hämtar som namnet anger nästa värde i ordningen hos en agent. På det här viset är det möjligt att stega sig igenom en lång rad värden fram tills det att noden, med ett felmeddelande, indikerar att det inte finns fler värden att hämta.

[5]Icke begärt agent-till-server-meddelande

set används för att sätta värdet på en variabel i en agent. Exempel-kommandon som kan utföras är:

- Stänga ner ett interface
- Nollställa ett register
- Logga ut en användare

trap används för att agenterna själva skall kunna sända information till den övervakande noden för att meddela om specifika händelser, utan att detta först har begärtst.

get bulk används för att meddela agenten att skicka "så mycket information den kan". Med det menas att den skall sända så mycket data som går på ett ordnat sätt. En del av datan kan finnas i speciella tabeller, formaterad på ett sådant sätt att det inte går att sända på ett begripligt sätt till den övervakande noden och därmed utesluts den ur sändningen.

Meddelanden som kan sändas över protokollet SNMP finns definierade i SNMP MIB[6]. Varje meddelande som kan sändas har sin unika sökväg inom denna MIB. Ett meddelande formateras enligt standarden ASN.1 som även används för exempelvis kryptering.

3.2 Fördelar

Det finns många fördelar med att kunna samla in och sammanställa data centralt. En av de främsta är att manuell administration av noderna inte behöver utföras separat då en uppgift skall exekveras på flera noder samtidigt, utan kan istället utföras från den gemensama punkten i nätverket. En annan fördel med SNMP är att man inte enbart har möjlighet att samla in data utan också utföra vissa enklare förändringar i konfigurationen hos agenten. Allt det här kan innefattas inom kategorin tidsbesparande fördelar.

[6]Management Information Base

Andra fördelar med SNMP är att det är ett standardiserat protokoll som är väl beprövat och som under en lång tidsperiod har utvecklats och testats för att uppnå maximal prestanda och en så hög säkerhetnivå som möjligt. SNMP är egentligen det enda protokollet som klarar av den uppgift det utför. Program som Netstat, ping med flera går visserligen att använda för att utföra en del av de uppgifter SNMP klarar av, men med sämre resultat. SNMP är ett flexibelt, plattformsoberoende och utbyggbart protokoll, vilket medger möjlighet att skapa egna delar av protokollet där anpassning för en viss typ av utrustning går att implementera.

3.3 Nackdelar

Trots att SNMP förenklar många av de steg som behöver göras vid övervakning av ett nätverk medför det inte enbart positiva effekter. I och med SNMPs långa utvecklingsperiod har protokollet vuxit sig stort, vilket i sin tur har inneburit att det blivit allt mer komplext. I dagsläget beräknas det därför ta omkring sex månader att sätta sig in i hur SNMP fungerar innan man kan börja dra nytta av protokollet och dess möjligheter[3]. En annan nackdel med SNMP är att om många noder i ett nätverk skall övervakas kan den redundanta datan i nätet bli stor på grund av alla paket protokollet skickar. Det i sin tur kan medföra att prestandan för den normala trafiken i nätverket försämras och istället för att bidra till effektivisering av nätverket får SNMP en omvänd effekt. En av anledningarna till att redundansen blir stor är uppbyggnaden av själva SNMP-paketen. Varje paket innehåller exempelvis information om vilken SNMP-version som “pratas”, samt utförlig information om var i MIB-trädet datan befinner sig.

4 Databasen

Databasen är den del av projektet där all data som samlats in från de övervakade noderna i nätverket lagras och även varifrån det grafiska användargränssnittet hämtar den information det presenterar. Databasen är knutpunkten mellan de två delprojekten och därmed en av de kritiska punkter som efter implementation och leverans kommer att vara svår att ändra, utan att stora delar av implementationen i det andra delprojektet behöver ändras. Då ingen form av data har samlats in i det här delprojektet, har databasen under utvecklingsperioden fyllts med "dummy-data" för att underlätta utvecklingen av de övriga delarna.

I kapitlet presenteras olika typer av relationer som förekommer i relationsdatabaser, samt hur de bör användas för att åstadkomma mer avancerade relationer mellan olika entiteter. De alternativa lösningar som diskuterades under utveckligen, men som inte ansågs uppfylla de krav som ställts diskuteras och presenteras i en egen avdelning i mitten av kapitlet. Slutligen presenteras de metoder och verktyg som användes vid implementationen. En mer fullständig beskrivning av implementationen och den slutliga designen av databasen finns beskriven i appendix C respektive appendix B.

Databasen innehåller informationen om de övervakade noderna som slutligen skall presenteras i gränssnittet gentemot användaren. Detaljer kring gränssnittet, samt hur selektionen av data sker från databasen, fördjupas i kapitel 5.

4.1 Relationsdatabaser

Relationsdatabaser är i dag det vanligaste sättet att designa en databas. I metoden används olika tabeller[7] för att lagra data som sedan knyts samman med relationer - därav ordet "realtionsdatabas". Det existerar tre olika typer av relationer; "en-till-en"-relationer, "en-till-många"-relationer samt "många-till-många"-relationer. Namnen på dessa relationer är

[7]Visualiseras som matrixer med olika fält såsom fältnamn, värden med mera

beskrivande för vad var och en av typerna gör.

en-till-en-relationer (figur 4.1) används för att knyta ett värde i en tabell till exakt ett annat värde i en annan tabell. Den här typen av relation används för att exempelvis ange att en person är chef på en avdelning på ett företag. Enbart en person kan här vara ytterst ansvarig och detta representeras då med den här typen av relation[4]. En person är dock enbart ansvarig på en avdelning i taget i det här fallet.

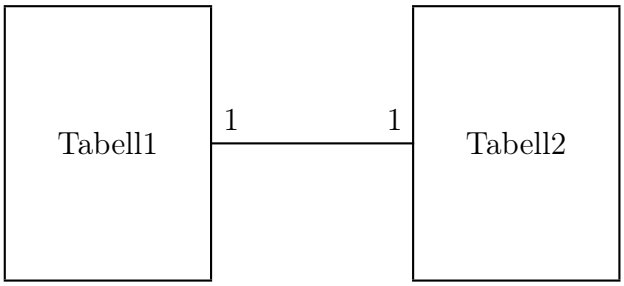

Figure 4.1: En-till-en-relation

en-till-många-relationer (figur 4.2) används för att knyta ett värde i en tabell till många olika värden i en annan tabell. Detta sker genom att man i tabell2 har ett fält som refererar till ett fält med unika värden i tabell1 (exempelvis primärnyckelfältet). Det här är den vanligaste typen av relation. Det är även två relationer av den här typen som används vid konstruktion av en många-till-många-relation.

många-till-många-relationer (figur 4.3) används för att kunna relatera många värden i en tabell till många värden i en annan. Relationen görs genom att skapa en ny tabell

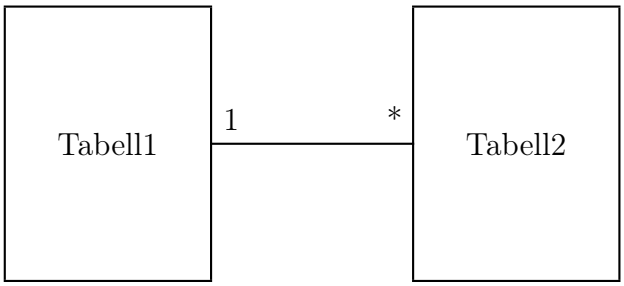

Figure 4.2: En-till-många-relation

"mellan" de två aktuella tabellerna. Den nya tabellen relateras i sin tur till de båda ordinarie med en-till-många-relationer. Ett exempel på användningsområden för den här typen av relation är när en anställd arbetar med flera projekt och varje projekt innehåller mer än en anställd.

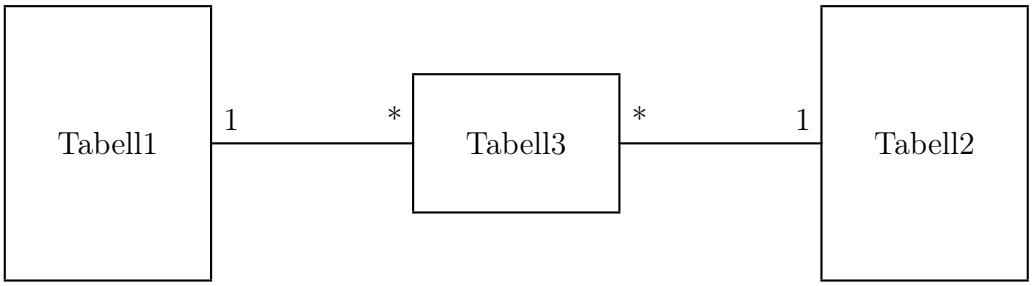

Figure 4.3: Många-till-många-relation

4.2 Designlösning

Databasens centrala roll har inneburit samarbete mellan båda grupperna, för att tillsammans komma överens om vilka data som bör samlas in och lagras om de olika typerna av utrustning som skall övervakas. Designen av databasen har skett enligt metoden för relationsdatabaser, då den metoden anses vara den mest effektiva för det här ändamålet och även den typ om vilken mest kännedom fanns. Vid designen lades arbete ner på att göra databasen så flexibel som möjligt, för att det i framtiden skall finnas möjlighet att bygga ut den för nya typer av utrustning som behöver övervakas.

För att åstadkomma den flexibilitet som eftersträvas separeras inte de olika typerna av utrustning som skall övervakas i egna entiteter, utan delas istället in i olika "typer" av utrustning. Anledningen till den här typen av indelning är att det i framtiden är möjligt att lägga till olika typer av utrustning, utan att behöva lägga till en helt ny separat del i databasen. Designmetoden möjliggjordes av att många av de data som samlas in om de olika typerna av utrustning är samma och därmed möjliga att lagra i samma entitet, tillsammans med en typindikation.

Övriga data som samlas in är separerade i olika entiteter - beroende på vilken typ av utrustning de lagrar data om. Entiteterna är sedan, antingen indirekt eller direkt kopplade till huvudentiteten "syss" (se tabell 4.1). En viss modifikation av databasen kan vara nödvändig vid införandet av nya typer av utrustning, beroende på hur pass inviduella de är. För en mer ingående beskrivning av designen samt hur de olika entiteterna är kopplade se appendix B.

Entiteten syss innehåller all den information som idag anses vara gemensam för alla typer av övervakad utrustning i nätverket. Därav kan det i framtiden vara nödvändigt att göra vissa förändringar i designen av databasen för att kunnan tillgodose de krav som ställs vid införandet av nya typer av övervakad utrustning.

Nyckel	Namn	Typ	Längd	Beskrivning
PK	sys_id	int	10	Systemets unika id
FK	sys_type nr	int	10	Refererar till sys_types:sys_type_id
	sys_sysName	varchar	30	Systemets unika namn i nätet
	sys_sysDescr	varchar	200	Beskrivning av systemet
	sys_sysContact	varchar	50	Kontaktperson för systemet
	sys_Location	varchar	30	Systemets fysiska position
	sys_UpTime	int	10	Systemets uptime
	sys_lastUpdateOp	int	11	Tidpunkt senast updaterad av systemansvarig <utime>
	sys_lastUpdateSys	int	11	Tidpunkt senast updaterad av systemet självt <utime>

Table 4.1: Entiteten syss

4.3 Alternativa design-lösningar

Ingen loggning på portar i hubbar och switchar

Nedan presenteras designen av den tabell som vid projektets början var tänkt att användas för lagring av den aktuella statusen på hubbarnas och switcharnas portar. Tanken var att först utesluta alla möjligheter till loggning av statusen på portarna och därmed spara plats i databasen. Då det visade sig nödvändigt att kunna övervaka statusen på portarna under vissa tidperioder gjordes en ny design som finns tillgänglig i appedix B. Den nya designen är inte implementerad utan enbart förberedd, men har för avsikt att implementeras i framtiden då systemet vidgas.

Nyckel	Namn	Typ	Längd	Beskrivning
PK	swh_prt_id	int	10	Portens unika id (UNIQUE)
KN	sys_nr	int	10	Refererar till syss:sys_id
	swh_prt_num	int	10	Portnummret i switchen/hubben
	swh_prt_speed	int	10	Porthastigheten i Mbit/se
	swh_prt_duplex	bool	10	Anger om porten arbetar med full duplex eller inte
	swh_prt_load	int	10	Belastningen på porten i kb/sek

Table 4.2: Den första versionen av entiteten swh_prts

Fördelen med den här typen av design är att lagringen av datan som samlas in blir betydligt enklare än vid den design som slutligen antogs. Nackdelen är att ingen form av statistik går att föra över hur pass hårt belastade portarna är och det är inte heller möjligt att i samma utsträckning göra justeringar i nätverket grundade på insamlad data.

4.4 Implementation

Implementationen för den databas där all insamlad data om de övervakade noderna lagras sker i det här projektet i mySQL[1]. MySQL lämpar sig bra för den här typen av projekt då den framförallt finns att tillgå för den Open Source-miljö som servern för övervakning skall köra i. Den uppfyller även de krav för prestanda och säkerhet som erfordras, samt att den samtidigt har möjlighet att använda sig av databasmotorn InnoDB för operationer mot databasen.

I den här delen av kapitlet presenteras de metoder och verktyg som används vid implementationen av databasen. Här diskuteras några fördelar med de typer av lagring som använts, samt en översiktlig historisk bakgrund till varför de kom till. Den teoretiska delen är främst till för dem som vill ha grundläggande kunskaper om det verktyg och metoder som användes vid implementationen. Personer som redan är insatta i de olika delarna kan överväga att hoppa över styckena.

4.4.1 MySQL

MySQL är ett databashanteringssystem som ursprungligen skapdes av Michael Widenius, men som idag utvecklas av det svenska företaget mySQL AB. Anledningen till att mySQL skapdes var att Widenius och de företag han samarbetade med var i stort behov av ett snabbt och flexiblet databashanteringssystem. Då de inte kunde hitta ett befintligt system som uppfyllde de krav de ställde, beslutade de att skapa ett eget[6].

MySQL är baserad på öppen källkod (Open Source), vilket är en av de främsta anledningarna till att mySQL idag är ett av de mest använda databashanteringssystemen på

marknaden. En annan anledning till framgången är Linux, som vid ungefär samma tidpunkt som mySQL började skaffa sig andelar på marknaden. Linux, som också är baserat på Open Source (och även en utmärkt runtime-miljö för mySQL), tillsammans med mySQL började förändra allmänhetens inställning till Open Source och istället få dem intresserade och se möjligheterna med den.

MySQL arbetar enligt Klient/Server (Client/Server) arkitekturen, där en separat server installeras, som sedan administreras med hjälp av klient-programvaror. Fördelen med att inte ha båda programvarorna i en gemensam applikation är att klienten körs betydligt mer sällan än servern och därför bara skulle ligga resistent i minnet och kräva prestanda av datorn, utan att egentligen utföra något. Nackdelen blir att klienten alltid är beroende av en server att ansluta sig mot samtidigt som servern inte går att administrera utan den (Moment 22).

4.4.2 InnoDB vs. myISAM

Introduktion

MySQL använder som standard en databasmotor som heter myISAM. MyISAM är mySQLs version av ISAM som IBM har utvecklat, men det finns flera andra alternativ. I det här projektet används en databasmotor vid namn InnoDB. Beskrivning och jämförelser följer nedan.

Fördelar

Den största fördelen med InnoDB är att den är transaktionsorienterad. Det innebär att om en operation av exempelvis insättningar i en databas avbryts, på grund av exempelvis en serverkrash, kommer inte delar av informationen att finnas kvar i databasen. Efter en krash går databasmotorn igenom tabellerna och jämför med de loggar som hela tiden skapas. Om en log inte stämmer överens med hur innehållet i databasen borde ha sett ut kommer en så kallad rollback att utföras och databasen "städas" därmed upp. En annan fördel med InnoDB är hastigheten. MyISAM är erkänt snabb, men InnoDB får bättre

resultat i de flesta prestandatester som görs. Nedan presenteras ett test i tabell 4.3 hämtat från http://www.innodb.com.

Operationer	**InnoDB**	**myISAM**
100000 INSERT	25 sekunder	40 sekunder
100000 SELECT på PK	57 sekunder	58 sekunder
100000 SELECT på SK	25 sekunder	95 sekunder

Table 4.3: Prestandatest av InnoDB vs. myISAM

5 GUI

I den här delen av projektet designas och implementeras det grafiska användargränssnittet där den insamlade informationen om de olika noderna i nätverket presenteras. Informationen hämtas ur den databas som skapades i kapitel 4. Som angivits tidigare i kravspecifikationen är målet att göra gränssnittet lättläst och strukturerat, främst för att en bredare målgrupp skall kunna förstå och arbeta med det. Gränssnittet implementeras i PHP i kombination med HTML, samt några JavaScript som kommer att användas för att öppna diverse nya popup-fönster.

Först presenteras tankar och idéer runt designen av gränssnittet, samt några tankegångar som bör övervägas för att göra gränssnittet så användarvänligt och lättförståeligt som möjligt. Förslag på olika design-tekniker ges, tillsammans med varför vissa av dem lämpar och inte lämpar sig för det här projektet. Slutligen presenteras en grundläggande översikt av PHP i implementations-delen av kapitlet, samt några för- och nackdelar det innebär att använda språket. I nästkommande kapitel sammanfattas resultatet av projektet samt de slutsatser som framkommit under utvecklingsperioden.

5.1 Design

Det grafiska användargränssnittet är en av de viktigaste delarna i en applikation, då det är genom gränssnittet användaren kommunicerar med den underliggande logiken. I och med gänssnittets centrala roll är det viktigt att användaren upplever designen som lättanvänd och välstrukturerad, med alla nödvändiga funktioner logiskt tillgängliga, samt nära till hands[7]. Gränssnitt där logik och struktur upplevs som svårförståeliga och ostrukturerade, tappar snabbt förtroende hos användaren och resulterar ofta i irritation. I vissa fall leder det även till att användaren slutar använda applikationen helt.

Det finns många bra metoder för att skapa ett gränssnitt gentemot den logik som existerar i grunden för applikationen. Signal-slot-metoden är en metod som används för

implementation av MVC[8]. MVC är vid design och implementation av ett program skrivet i till exempel C++, ett mycket effektivt sätt att separera logiken från gränssnittet, men samtidigt behålla en nära kommunikation mellan dem[8]. Då den här applikationen är skriven i PHP försvårar PHPs arkitektur möjligheten till användning av MVC, eftersom PHP bland annat inte stöder varken signals eller slots. Beslut fattades istället om att i PHP-filerna i största möjliga utsträckning separera den kod som utför logiska operationer, från den som genererar delar av gränssnittet.

5.2 Att tänka på vid design av grafiska användargränssnitt

Det finns ett antal frågor som bör ställas innan designen av ett användargränssnitt påbörjas. Nedan presenteras och besvaras de frågeställningar som under utvecklingen visade sig vara de viktigaste vid design av gränssnitt.

5.2.1 Frågeställningar

- För vem designas gränssnittet?

- Vilken utbildning har användaren?

- Vad skall göras och varför?

- Vad är nödvändiga effekter?

[8]Model View Control

För vem designas gränssnittet?

Vad alla som designar ett gränssnitt alltid bör ha i åtanke är att det är för användaren gränssnittet designas. Som programmerare är det lätt att stirra sig blind och tycka att upplägget i gränssnittet är självklart och logiskt. Användaren har i många fall en helt annan syn på upplägget och finner det i själva verket krångligt och ologiskt. Det är därför viktigt att vid designen hela tiden ta hänsyn till vad de slutliga användarna har för åsikter och i största möjliga utsträckning försöka tillgodose alla krav och förändringar de kommer med.

Vilken utbildning har användaren?

Lika viktigt som att inte lägga sig på en för hög nivå gentemot användaren, är det att inte lägga sig på en för låg nivå. Att designa ett gränssnitt som språkmässigt och strukturmässigt ligger på samma kunskapsnivå som användaren, skapar högre arbetsmoral och gör att användaren behöver anstränga sig mindre.

Vad skall göras och varför?

Vid designen av ett gränssnitt är det viktigt att ha en klar bild av vad som skall utföras i gränssnittet och varför. Utan svaren till de två frågorna är det svårt att skapa ett gränssnitt som är anpassat efter den uppgift som skall utföras. Design-teamet bör i vissa fall även konsultera med slutanvändarna och gemensamt med dem komma fram till en lösning.

Vad är nödvändiga effekter?

En annan sak att tänka på är att inte överarbeta gränssnittet, utan göra det enkelt och funktionellt. Visst är det snyggt med en extra animation på knapparna, exempelvis en pil som blinkar och pekar på en dörr för att indikera var man stänger av programmet. Frågan är bara om det fyller någon funktion för användaren, eller om det i själva verket kan vara något som distraherar användaren och belastar hårdvaran för mycket? Fyller inte en del av gränssnittet någon funktion för användaren bör den tas bort eller omarbetas.

5.2.2 Applicering av frågeställningar på projektet

I den här delen av kapitlet besvaras frågorna ställda i föregående delkapitel gällande hur beslut har fattats för att på bästa möjliga sätt tillgodose de svar som givits.

För vem designas gränssnittet?

Som nämnts tidigare är ett av målen med projektet att skapa en programvara som är både lättförståelig och lättöverskådligt. Gränssnittet har därför ingen speciell målgrupp som tänkt slutanvändare, utan riktar sig till alla som är i behov av den här typen av programvara.

Vilket utbildning har användaren?

Slutanvändaren behöver besitta grundläggande kunskaper inom datoranvändande och internet. Språket i gränssnittet är skrivet på en engelsk, mycket låg nivå och i nästan helt utan tekniskt avancerade termer. Kunskapskraven kan därför jämföras med vanlig gymnasienivå i engelska. Installationen av programvaran behöver djupare kunskaper inom Debian Linux och administration av webserver. Konfiguration av serversida i PHP[5] är även i vissa fall nödvändig och kan därför räknas som ett av kunskapskraven inför installationen.

Vad skall göras och varför

Då det stod klart redan innan designen av gränssnittet påbörjades, vad som skulle göras i det och varför, ansågs designen ganska okomplicerad. Då nästan ingen interagering med gänssnittet kan ske behövdes heller inga komplicerade användarfall analyseras och utvärderas. Se kravspecifikationen i kapitel 2.

Vad är nödvändiga effekter?

Gränssnittet är i stort sett helt utan effekter och animationer, för att bibehålla stabilitet och snabbhet och möjliggöra för användare med modem- eller GPRS-uppkopplingar att kunna övervaka.

5.3 Objekt

Med objekt menas alla de former av komponenter (till exempel knappar, texter, grafer, m.m. (eng. widgets)) som placeras ut i gränssnittet för att representera eller presentera olika former av data eller information. Vid användning av olika typer objekt då ett gränssnitt designas måste stor försiktighet vidtagas. Användning av fel objekt kan göra användaren förvirrad, vilket kan leda till oönskade resultat vid de operationer som utförs. Nedan följer ett par exempel på riktlinjer som är bra att tänka på vid placering av objekt i ett gränssnitt. Applicering av riktlinjerna på projektet presenteras i den andra delen av kapitlet.

5.3.1 Riktlinjer

Rätt typ

Det är viktigt att välja rätt typ av objekt när man representerar eller presenterar olika typer av data eller information. Om ett objekt uppfattas fel av användaren kan det leda till mer distraktion än hjälp. I värsta fall utförs en helt annan operation än vad användaren hade för avsikt att utföra, vilket i sin tur kan leda till stora förluster av data, eller annan form av fel.

Konsekvens

Användningen av en viss typ av objekt i ett gränssnitt måste göras konsekvent, det vill säga samma typ av objekt måste användas för att presentera samma typ av information eller för att utföra samma typ av operation överallt. Om samma typ av objekt används för att representera olika saker vid olika tillfällen kan användaren blir förvirrad och är tvungen att ägna onödig tid åt att sätta sig in i vad objektet, vid det aktuella tillfället, representerar.

Storlek

Storleken på objekt har en avgörande betydelse för hur användaren reagerar. Stora objekt drar till sig större uppmärksamhet än små och bör därför användas med försiktighet, eller

enbart då det är nödvändigt att göra användaren extra uppmärksamm på att en viss typ av händelse har inträffat. Ett objekt bör inte vara större än "vad nöden kräver" och om flera objekt av samma typ representerar samma typ av data eller funktion bör de ha samma storlek, där objektet som kräver mest utrymme bestämmer storleken för de resterande.

Färgsättning

Färgsättningen på objekten är ett av de viktigaste verktygen för att upplysa en användare om att något har skett, eller visa att statusen för ett objekt har ändrats. Det är viktigt att välja rätt färger för att upplysa en användare om vilken typ[9] av meddelande han eller hon konfronteras med. Desto lättare det är för användaren att identifiera ett meddelande, desto kortare tid tar det att åtgärda eventuella fel.

Färger för att indikera de olika typerna av meddelanden varierar från kultur till kultur. Ett felmeddelande i de västerländska länderna indikeras oftast med att objektet eller meddelandet får eller byter färg till röd, medan samma meddelande kan misstolkas i exempelvis Kina, där färgen röd betyder lycka.

För att möjliggöra för färgblinda att kunna använda och förstå gränssnittet bör inte färgerna röd och grön ensamma användas för att indikera olika status på samma objekt. Måste färgerna användas bör det göras i kombination med någon annan indikation - till exempel ett textmeddelande.

Placering

Placeringen av objekten i ett gränssnitt är avgörande för användarvänligheten. Undersökningar har visat att man med ögonen börjar söka av skärmen efter information i det övre vänstra hörnet, för att sedan fortsätta i en klockcykel runt och avsluta i det nedre vänstra. Den här metoden innebär att om viktig och kritisk information placeras nere i vänstra hörnet av gränssnittet tar det lång tid innan användaren uppfattar vad som har hänt och kan reagera. Reaktionstiden för händelser förlängs därmed avsevärt mot vad den hade kunnat vara om informationen hade placerats uppe till vänster i gränssnittet. För att und-

[9] varning, error, m.m.

vika syndromet skall alltså information som kräver omedelbar interaktion med användaren placeras uppe i det vänstra delen av gränssnittet medan information som bara informerar om mindre kritiska saker kan placeras i de övriga delarna, beroende på prioritering[8].

5.3.2 Applicering av riktlinjer

Rätt typ

För att inte förvilla användaren har de typer av objekt som används i gränssnittet följt den standard som anses vara allmän - det vill säga: En rubrik är alltid enbart en rubrik och har inte till exempel funktionen av en länk. Hänsyn har tagits till att göra objekten så självförklarande som möjligt.

Konsekvens

Alla objekt har använts på ett konsekvent och korrekt sätt och utför alltid samma typ av uppgift. Ett bra exempel är namnen på de övervakade noderna, som alltid går att klicka på för att visa den mer detaljerade informationen.

Storlek

Det objekt som tillägnats den största individuella platsen i gränssnittet är diagrammet för övervakning av utrymmet på servrarnas hårddiskar. Den huvudsakliga anledningen till det är att servrarnas hårddiskar är kritiska punkter för de tjänster som körs i nätverket och att en diskkrash e orsaka ett stort antal timmars arbete som lätt hade kunnat undvikas. För en närmare skiss över gränssnittet, se appendix E.

Färgsättning

För att underlätta för röd-grön-färgblinda byter de varningar som visas färg till röd (eller gul, beroende på typ), samt kombineras med ändring av den font som visar meddelandet. En text är dock aldrig grön. Bakgrunden i gränssnittet har satts till en mörkare grå ton för mjukare beskådning för ögonen. Den gröna färgen som används som bakgrund till rubrikerna är företagets egna gröna färg.

Placering

Då servrar ansågs vara den delen av ett nätverk som var mest kritisk, placerades dessa i den övre vänstra delen av gränssnittet. Det näst mest kritiska ansågs vara skrivare och hubbar som placerades i den övre högra delen av gränssnittet följt av arbetsstationer i den nedre. Diagrammet som övervakar servrarnas hårddiskar placerades i den nedre vänsta delen av gränssnittet (som anses ha den lägsta prioriteten gällande tid det tar att upptäcka informationen), men det kompenserades genom den större individuella platsen i gränssnittet.

5.4 Skärmdisposition och gränssnittsförslag

Lika viktigt som att använda rätt storlek, färg och placering på objekten är disponeringen av gränssnittets yta vid designen. Har man en liten skärm innebär det att ytan måste disponeras mer sparsamt och informationen kan behövas presenteras mer kompakt. En stor skärm innebär möjligheten att presentera informationen luftigare och mer lättläst, men innebär samtidigt risken att allt för mycket information presenteras på samma gång, vilket leder till att informationen lätt blir svåröverskådlig.

Under designfasen av gränssnittet gjordes flera olika förslag och metoder på hur layout och funktioner skall presenteras och placeras. En diskussion följer här om vilka för- och nackdelar respektive metod medför. Utgångspunkten för gränssnittet är de fyra huvudkategorier av utrustning som skall övervakas. Att switchar och hubbar skulle presenteras tillsammans i en gemensam del av gränssnittet var ett av de designbeslut som fattades först. När beslut om hur skärmytan skulle disponeras mellan de fyra utrustningskategorierna arbetade vi fram följande alternativ till design.

Varje ruta är så stor att datan får plats i den

Det här alternativet innebär att fönsterna får den plats de behöver och "scrollning" behöver utföras om en annan status skall överskådas. Den största nackdelen med en här metoden är att inte någon ordentlig översikt över skärmen åstadkoms. Lösningen skall enligt specifikation vara lättöverskådlig, alltså är inte "scrollning" någon bra idé.

Alla rutor är lika stora

Den här designmetoden innebär att alla de fyra olika ramarna görs lika stora för att på så sätt öka symmetrin i gränssnittet. Nackdelen är att inte alla rutor utnyttjas optimalt och delar av skärmen blir outnyttjad. Då skärmytan redan är begränsad är det av stor vikt att all yta används optimalt, men utan att förlora överblicken över gränssnittet. Den här lösningen var först den metoden som var tänkt att implementeras, men då behovet av den outnyttjade ytan uppstod, fattades beslut om att designa om gränssnittet.

Informationen presenteras här i åtta olika ramar där varje del har två ramar var. En för information om datorer samt en för de händelser som triggas.

servrar info	skrivare info
servrar händelser	skrivare händelser
hubbar/switchar info	workstations info
hubbar/switchar händelser	workstations händelser

Figure 5.1: Alla rutor är lika stora

Den slutliga designen

Ett mellanting av de två först beskrivna lösningarna var till slut den som fastslogs som den mest lämpliga designen av gränssnittet i det här projektet. De olika fönsterna får här olika storlek beroende på hur mycket information som uppskattas behövs i varje del av gränssnit-

tet. Den stora fördelen är möjligheten att utnyttja hela skärmen optimalt, men förlusten av symmetrin gör sökning efter information i fönsterna långsammare. Designlösningen går ut på att serverövervakningen får mer plats i gränssnittet än övriga då den anses vara den viktigaste delen att övervaka. Samtidigt tilldelas hubbar och switchar endast ett gemensamt fönster, som i de andra fallen är avsett för händelser, då det inte uppkommer några händelser från en switch eller hubb utan enbart statistiska fakta om enheten i sig. Istället tilldelas serverövervakningen ett diagram för möjlighet till övervakning av utrymmet på de hårddiskar som sitter i servrarna. Delen för övervakning av arbetstationer (workstations) förblir oförändrad. Det här är den lösningen vi valt att implementera i projektet då bland annat servrar kräver mer utrymme i gränssnittet än exempelvis switchar. För en detaljerad skiss samt skärmdumpar över den slutliga designen se appendix E och D.

<table>
<tr><td rowspan="2">servrar
info</td><td>skrivare
info</td></tr>
<tr><td>skrivare
händelser</td></tr>
<tr><td>servrar
händelser</td><td>hubbar/switchar
info</td></tr>
<tr><td>servrar
diagram</td><td>workstations
info</td></tr>
<tr><td>allmän
information</td><td>workstations
händelser</td></tr>
</table>

Figure 5.2: Den slutliga designen

5.5 Implementation

I den här delen av kapitlet diskuteras teknikerna som används vid implementationen av gränssnittet. Som tidigare nämnts användes PHP[10], för att implementera logiken i gränssnittet och för att hämta information om de övervakade noderna i nätverket från databasen. Implementationen av gränssnittet är den i huvudsak mest praktiska delen i projektet. Då det i stort sett bara är programkod som skrivs presenteras endast en översiktlig teoretisk beskrivning av vilka verktyg och metoder som används. För personer med redan goda kunskaper inom PHP och dess struktur kan den här delen av kapitlet läsas "kursivt". För en närmare presentation av implementationen se appendix G.

PHP

PHP är ett serverbaserat, plattformsoberoende skriptspråk[9]. Språket är besläktat med C, vilket medför att folk som tidigare programmerat och arbetat med C ofta har har lätt att lära sig PHP. Syntaxen i de båda språken är i stort sett den samma. Den huvudsakliga skillnaden mellan dem är att det i PHP är möjligt att skapa klasser och objekt som innebär möjlighet till objektorienterad programmering. PHP är inget standardiserat språk. Det har medfört att funktioner och parametrar har ändrats mellan versionerna, vilket kräver att applikationerna har behövt skrivas om och testats igenom med den nya versionen.

Med "serverbaserat" menas att all PHP-kod exekveras på servern och att resultatet av exekveringen sedan skickas till klienten för presentation. Den här metoden skiljer sig från till exempel vanlig HTML där sidan i sin helhet skickas till klienten, som sedan presenterar den.

Med "plattformoberoende" menas att PHP kan köras på nästan vilken plattform[11] som helst, utan att man först behöver anpassa koden. Modulen för att kunna exekvera PHP-kod är givetvis anpassad för den specifika plattform den körs på, men det API[12] som presenteras gentemot programmeraren, är det samma på alla plattformar.

[10] PHP: Hyptertext Processor
[11] Unix, Linux, Windows, Mac, etcetera
[12] Application Programming Interface

PHPs skripspråksegenskap innebär att det är ett interpreterande språk, det vill säga koden behöver inte kompileras innan exekvering. Intepreterande språk är oftast lättare att arbeta med, då kompilering efter varje ändring inte behövs för att se de förändringar som gjorts i koden. Nackdelen med dem är att exekveringen av koden oftast tar längre tid, då koden måste översättas till maskinkod i realtid varje gång den körs för att sedan kunna exekveras.

Punktlista - Fördelar med PHP

- Nära besläktat med C
- Serverbaserat
- Plattformsoberoende

Punktlista - Nackdelar med PHP

- Interpreterande
- Ändringar i språket
- Ej standardiserat

6 Sammanfattande kommentarer

För att sammanfatta alla föregående kapitel och komma fram till någon form av slutsatser görs detta lättast genom att dela in dessa i teoretiska och praktiska. Tillsammans har de två delarna tidsmässigt varit väl avvägda och tidsplanen har kunnat följas utan större problem eller förseningar.

6.1 Designdelen

Den teoretiska delen av projektet har tagit upp den största delen av tiden. För att göra resterande arbete i de båda delprojekten så lätt som möjligt påbörjades designen av databasen redan efter cirka en vecka. Databasens design var tvungen att vara flexibel och lätt att modifiera, då det i framtiden skulle vara möjligt att ändra i den för kommande typer av övervakad utrustning. Därför resulterade designen i en "gemensam" tabell där generell information om alla typer av övervakad utrustning kunde samlas in och därifrån vidare kopplat till entiteter med specifik information som bara samlades in om vissa typer av utrustning.

Projektet skall enligt kravspecifikationen (se kapitel 2) vara anpassat för en så bred målgrupp som möjligt och därmed vara enkelt och lättanvänt. Därför ägnades mycket tid åt att ta reda på de olika kriterier och tekniker som var möjliga att senare applicera på den praktiska delen vid design av gränssnittet. Den slutliga designen resulterade i den som står beskriven i kapitel 5.

6.2 Implementationsdelen

Den praktiska delen av projektet har helt varit koncentrerad på implementation av databas och gränssnitt. Implementationen av databasen gjordes på cirka en arbetsdag och gränssnittet slutfördes på ungefär tre veckor från det att det påbörjades. Då det var en relativt lätt uppgift att implementera fortskred arbetet ganska snabbt och inga större problem

uppstod.

Sammanfattningsvis kan det konstateras att det har varit ett mycket roligt och lärorikt projekt där en del av de situationer som uppkommer i arbetslivet har fått erfaras. Slutsatsen blir att det kommer att bli intressant att följa upp arbetet med produkten för att se hur andra användare upplever den .

References

[1] MySQL AB. *MySQL: The World's Mos Popular Open Source Database.* http://www.mysql.org, 2003-12-01.

[2] Inc. Allen Systems Group. *An Overview Of SNMP Protocol.* Internet, 2002.

[3] Thomas R. Cikoski. *Simple Network Management Protocol, part 1 of 2.* UseNet, 2003.

[4] Shamkant B. Elmasri, Ramez & Navathe. *Fundamentals of Database Systems 3rd edition.* Addison Wesley, 1997.

[5] The PHP Group. *PHP: Hypertext Preprocessor.* http://www.php.net, 2003-12-01.

[6] Mark Maslakowski. *Lär dig MySQL på 3 veckor.* SAMS, 2000.

[7] Hannes Persson. *Kurskompendium Grafiska anvädargrässnitt DAV B05 Föreläsningar.* KAU, 2002.

[8] Hannes Persson. *Kurskompendium Grafiska anvädargrässnitt DAV B05 Kurslittertur.* KAU, 2002.

[9] Larry Ullman. *PHP Visuell Snabbguide.* Peachpit Press, 2002.

A Förkortningar

ASN.1	Abstract Syntax notation one
GUI	Graphical User Interface
ISAM	Indexed Sequential Access Method
MIB	Management Information Base
RFC	Request For Comments
SNMP	Simple Network Management Protocol
SQL	Simple Query Language
PHP	PHP: Hypertext Processor
SQL	Structured Query Language
SSH	Secure SHell
MVC	Model View Control
HTML	HyperText Markup Language
GPRS	General Packet Radio Services
API	Application Programming Interface
myISAM	mySQLs implementation av ISAM
E-R	Entity - Relation
TCP	Transmission Control Protocol
CPU	Central Processing Unit
RAM	Random Access Memory
ASCII	General Packet Radio Services
UDP	User Datagram Protocol

B Databasbeskrivning

I det här kapitlet förklaras de olika delarna av databasen, samt med grafiska skisser hur de är relaterade till varandra. Alla entiteter som ingår i databasen presenteras och förklaras mer ingående. Tillhörande attribut beskrivs i detalj, gällande typ, längd och eventuell uppgift det har.

Två grafiska skisser över hur entiteterna är relaterade till varandra återfinns i slutet av kapitlet. Den första är ett relationsdiagram där pilar mellan de olika entiteterna representerar en-till-många relationer. Den andra grafen är ett ER-diagram för lite mer detaljerad beskrivning av relationerna mellan entiteterna, samt för att klargöra namnen på de olika relationerna.

B.1 Beskrivning av entiteter

- **syss** - Här lagras de data som är genemsamma för alla typer av utrustning som övervakas.
- **sys_nics** - Här lagars data om de olika NICs som är relaterade till systemen.
- **sys_ips** - Här lagras data om de olika IP-adresser som är relaterade till nicsen.
- **sys_types** - Här lagras namnen på de olika typerna av utrustning som går att övervaka.
- **prn_es** - Här lagras data om de olika events som kommer från skrivarna.

- **prn_hrDevicsStatus** - Här lagras de olika typer av status som en skrivare kan ha, relaterade till skrivar-eventsen.

- **prn_hrPrinterStatus** - Här lagras de olika typer av status som utskriftdelen i en skrivare kan ha, relaterade till skrivar-eventsen.

- **prn_hrPrinterDetectedErrorState** - Här lagras de olika typer av felmeddelande som en skrivare kan generera, relaterade till skrivar-eventsen.

- **swh_prts** - Här lagras data om de olika portarna som är relaterade till en hubb/switch.

- **swh_prt_es** - Här lagras data om belastning på de olika hubbarnas/switcharnas portar är vid en bestämd tidpunkt.

- **comp_hdds** - Här lagras data om de olika hårddiskarna som är relaterade till en viss arbetstation eller server.

- **comp_es** - Här lagras data om olika status hos arbetsstationen/servern vid en bestämd tidpunkt.

B.2 Specificering av entiteter

B.2.1 syss

Nyckel	Namn	Typ	Längd	Beskrivning
PK	sys_id	int	10	Systemets unika id
FK	sys_type nr	int	10	Refererar till sys_types:sys_type_id
	sys_sysName	varchar	30	Systemets unika namn i nätet
	sys_sysDescr	varchar	200	Beskrivning av systemet
	sys_sysContact	varchar	50	Kontaktperson för systemet
	sys_Location	varchar	30	Systemets fysiska position
	sys_UpTime	int	10	Systemets uptime
	sys_lastUpdateOp	int	11	Tidpunkt senast updaterad av systemansvarig <utime>
	sys_lastUpdateSys	int	11	Tidpunkt senast updaterad av systemet självt <utime>

B.2.2 sys_types

Nyckel	Namn	Typ	Längd	Beskrivning
PK	sys_type_id	int	10	Systemtypens unika id
	sys_type_descr	varchar	20	Anger typ av sysem

B.2.3 sys_nics

Nyckel	Namn	Typ	Längd	Beskrivning
PK	sys_nic_id	int	10	NICets unika id
FK	sys_nr	int	10	Refererar till syss:sys_id
	sys_nic_PhysAddress	char	17	NICets MAC-adress

B.2.4 sys_ips

Nyckel	Namn	Typ	Längd	Beskrivning
PK	sys_ip_id	int	10	IP-Adressens unika id
FK	sys_nic_nr	int	10	Refererar till sys_nics:sys_nic_id
	sys_ip_NetAddress	varchar	15	En av IP-Adresserna som tillhör NICet

B.2.5 prn_es

Nyckel	Namn	Typ	Längd	Beskrivning
PK	prn_e_id	int	10	Printer-eventets unika id
FK	sys_nr	int	10	Refererar till syss:sys_id
FK	prn_e_hrDevice Status_nr	int	2	Refererar till prn_hrDeviceStatuss prn_hrDeviceStatus_id
FK	prn_e_hrPrinter Status_nr	int	2	Refererar till prn_hrPrinterStatuss:prn_hrPrinterStatus_id
FK	prn_e_hrPrinterDetectedError State_nr	int	2	Refererar till prn_hrPrinterDetectedErrorStates:prn_hrPrinterDetectedErrorStates_id
	prn_e_date	int	11	Tidpunkten då eventet skedde <utime>

B.2.6 prn_hrDeviceStatuss

Nyckel	Namn	Typ	Längd	Beskrivning
PK	prn_hrDevice Status_id	int	2	PrinterDeviceStatusets unika id
	prn_hrDevice Status_descr	varchar	20	PrinterDeviceStatusets beskrivning

B.2.7 prn_hrPrinterStatuss

Nyckel	Namn	Typ	Längd	Beskrivning
PK	prn_hrPrinter Status_id	int	2	Printerstatusens unika id
	prn_hrPrinter Status_descr	varchar	20	Printerstatusens beskrivning

B.2.8 prn_hrPrinterDetectedErrorStates

Nyckel	Namn	Typ	Längd	Beskrivning
PK	prn_hrPrinter DetectedError State_id	int	2	PrinterDetectedErrorStatsets unika id
	prn_hrPrinter DetectedError State_descr	varchar	20	PrinterDetectedErrorStatsets beskrivning

B.2.9 swh_prts

Nyckel	Namn	Typ	Längd	Beskrivning
PK	swh_prt_id	int	10	Portens unika id (UNIQUE, men inte PK)
	sys_nr	int	10	Refererar till syss:sys_id
	swh_prt_num	int	10	Portnummret i switchen/hubben
	swh_prt_speed	int	10	Porthastigheten i Mbit/se
	swh_prt_duplex	int	1	Anger om porten arbetar med full duplex eller inte

B.2.10 swh_prt_es

Nyckel	Namn	Typ	Längd	Beskrivning
PK	swh_prt_e_id	int	10	Eventets unika id
FK	swh_prt_nr	int	10	Refererar till swh_prts:swh_prt_id
	swh_prt_e_date	int	11	Tidpunkten då eventet inträffade <utime>
	swh_prt_e_load	int	10	Belastningen på porten i kb/sek

B.2.11 comp_hdds

Nyckel	Namn	Typ	Längd	Beskrivning
PK	comp_hdd_id	int	10	Hårddiskens unika id
FK	sys_nr	int	10	Refererar till syss:sys_id
	comp_hdd_size	float	10	Hårddiskens storlek angivet i Gigabyte med en decimal
	comp_hdd_left	float	10	Utrymme kvar på hårddisken angivet i Gigabyte, med en decimal
	comp_hdd_mpoint	varchar	30	Hårddiskens mountpoint i systemet
	comp_hdd_peak	float	10	Maximalt med data som någonsin varit lagrat på disken angivet i Gigabyte, med en decimal
	comp_hdd_ peak_date	int	11	Tidpunkten då den maximala datan var lagrad <utime>

B.2.12 comp_es

Nyckel	Namn	Typ	Längd	Beskrivning
PK	comp_e_id	int	10	Eventets unika id
FK	sys_nr	int	10	Refererar till syss:sys_id
	comp_e_temp	int	10	Systemets temeratur då mätningen gjordes
	comp_e_memusage	int	10	Systemets användning av minnet i % då mätningen gjordes
	comp_e_cpuusage	int	10	Systemets användning av cpu:n i % då mätningen gjordes
	comp_e_date	int	11	Tidpunkten då mätningen gjordes

B.3 Förkortningar i databasen

swh	Switchar och hubbar
prn	Printer
e(s)	Event(s)
prt	Port
id	Primärnyckel
nr	Nummer (referens för främmandenyckel)
PK	Primary Key (Primärnyckel)
FK	Foreign Key (Främmandenyckel)
num	Löpande nummer
sys	System
op	Operatör/Administratör
hdd	Hårddisk
comp	Computer/Dator (Server/Arbetsstation)
ip	IP-Adress
temp	Temperatur
nic	Network Interface Card
mpoint	Mountpoint
mem	Memory (Minne)
cpu	Central Processing Unit (Processor)
us	Unsigned

B.4 ER-Diagram

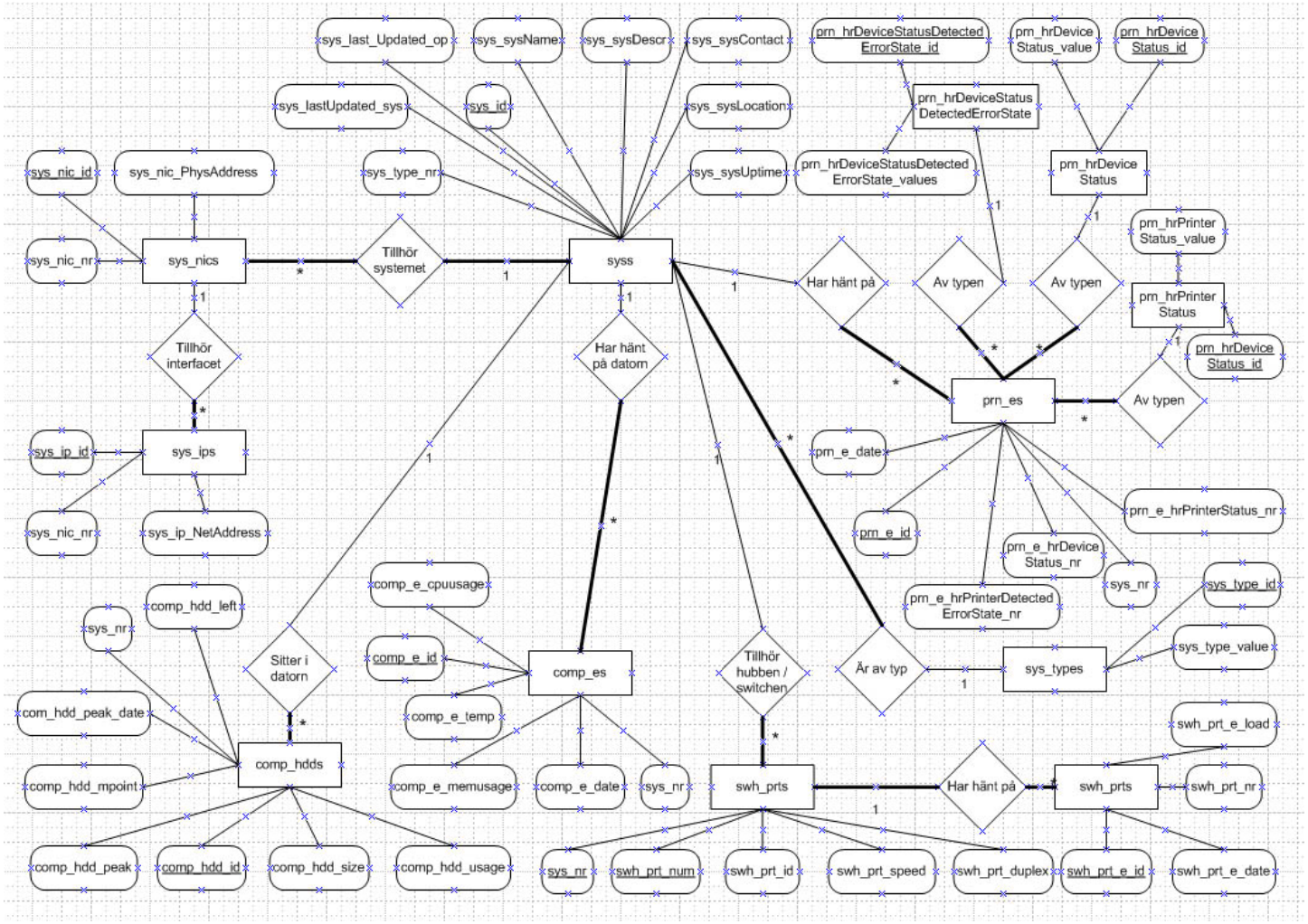

Figure B.1: ER-Diagam

B.5 Databastabeller

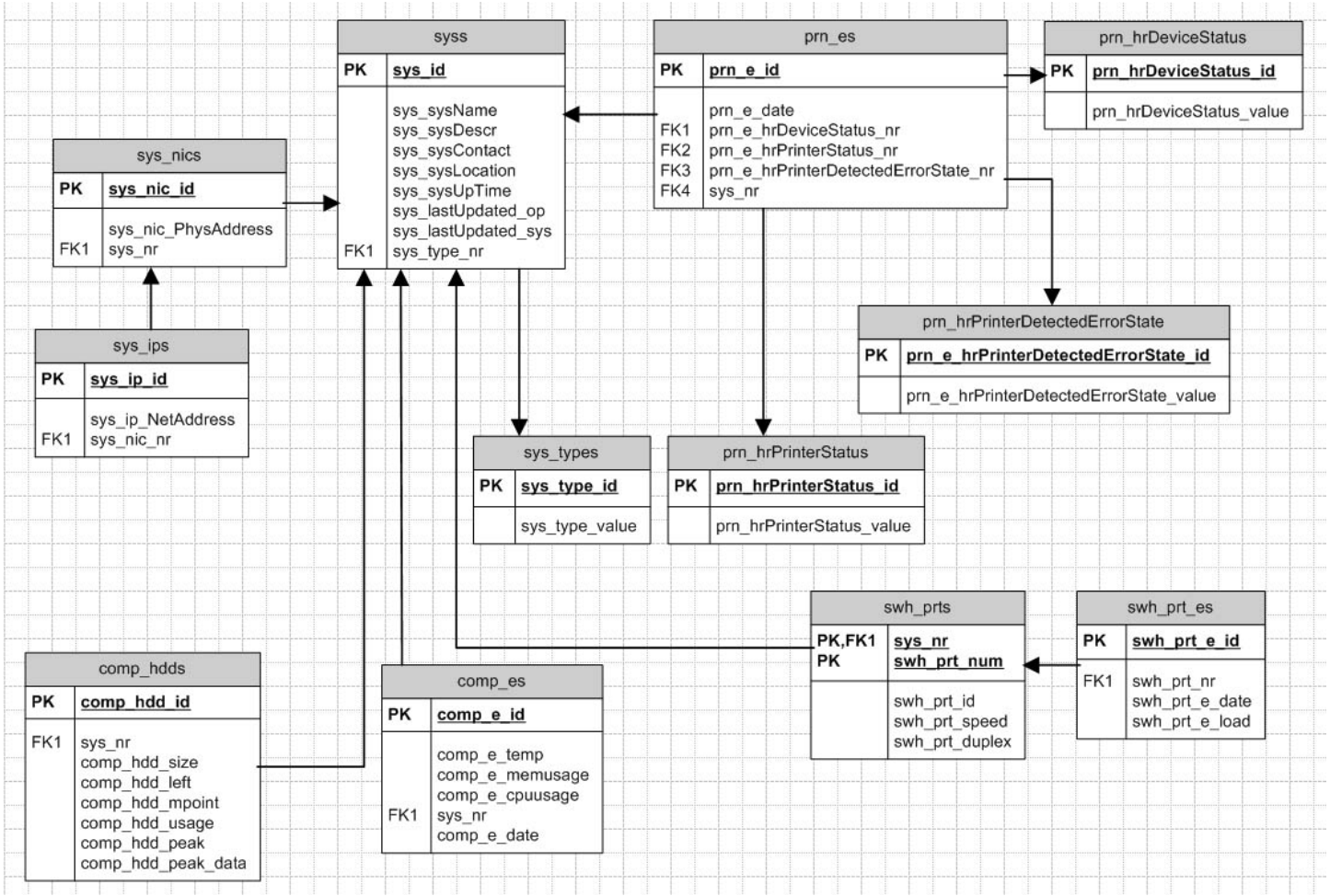

Figure B.2: Databastabeller

C Databasimplementation

C.1 Implementationsmetod

För att implementera och initiera databasen har nedanstående script skapats för att underlätta och effektivisera arbetet. De värden och tabeller som har ansetts statiska är implementerade och initierade från början, för att underlätta vid eventuell nollställning av databasen. Scriptet innehåller exakt alla steg i implementationen och står även i den ordning de utförs vid körning.

C.2 Implementations-script

```
CREATE DATABASE snmpdata;
use snmpdata;

CREATE Table sys_types(
  sys_type_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_type_descr varchar(20)
)type=InnoDB;

CREATE Table syss(
  sys_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_sysName varchar(30),
  sys_sysDescr varchar(200),
  sys_sysContact varchar(50),
  sys_sysLocation varchar(30),
  sys_sysUpTime int(10),
  sys_lastUpdate_op int(11),
  sys_lastUpdate_sys int(11),
  sys_type_nr int(10)#,
  # FOREIGN KEY (sys_type_nr) REFERENCES sys_types(sys_type_id)
)type=InnoDB;

CREATE Table sys_nics(
  sys_nic_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_nr int(10),
  sys_nic_PhysAddress char(17)#,
  # FOREIGN KEY (sys_nr) REFERENCES syss(sys_id)
)type=InnoDB;

CREATE Table sys_ips(
  sys_ip_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```
  sys_nic_nr int(10),
  sys_ip_NetAddress varchar(15)#,
  # FOREIGN KEY (sys_nic_nr) REFERENCES sys_nics(sys_nic_id)
)type=InnoDB;

CREATE Table prn_es(
  prn_e_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_nr int(10),   prn_e_hrDeviceStatus_nr int(2),
  prn_e_hrPrinterStatus_nr int(2),
  prn_e_hrPrinterDetectedErrorState_nr int(2),
  prn_e_date int(11)#,
  # FOREIGN KEY (sys_nr) REFERENCES sys(sys_id),
  # FOREIGN KEY (prn_e_hrDeviceStatus_nr) REFERENCES
prn_hrDeviceStatuss(prn_hrDeviceStatus_id),
  # FOREIGN KEY (prn_e_hrPrinterStatus_nr) REFERENCES
prn_hrPrinterStatuss(prn_hrPrinterStatus_id),
  # FOREIGN KEY (prn_e_hrPrinterDetectedErrorState_nr) REFERENCES
prn_hrPrinterDetectedErrorStates(prn_hrPrinterDetectedErrorState_id)
)type=InnoDB;

CREATE Table prn_hrDeviceStatuss(
  prn_hrDeviceStatus_id int(2) NOT NULL PRIMARY KEY,
  prn_hrDeviceStatus_descr varchar(20)
)type=InnoDB;

CREATE Table prn_hrPrinterStatuss(
  prn_hrPrinterStatus_id int(2) NOT NULL PRIMARY KEY,
  prn_hrPrinterStatus_descr varchar(20)
)type=InnoDB;

CREATE Table prn_hrPrinterDetectedErrorStates(
  prn_hrPrinterDetectedErrorState_id int(2) NOT NULL PRIMARY KEY,
  prn_hrPrinterDetectedErrorState_descr varchar(20)
)type=InnoDB;

CREATE Table swh_prts(
  swh_prt_id int(10) NOT NULL PRIMARY KEY,
  sys_nr int(10),
  swh_prt_num int(10) NOT NULL,
  swh_prt_speed int(10)#,
  # FOREIGN KEY (sys_nr) REFERENCES syss(sys_id)
)type=InnoDB;

ALTER Table swh_prts DROP PRIMARY KEY;
ALTER Table swh_prts ADD PRIMARY KEY (swh_prt_id, swh_prt_num);
ALTER TABLE swh_prts CHANGE COLUMN swh_prt_id swh_prt_id int(10) NOT NULL
AUTO_INCREMENT;

CREATE Table swh_prt_es(
```

```
  swh_prt_e_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  swh_prt_nr int(10),
  swh_prt_e_date int(11),
  swh_prt_e_load int(10)#,
  # FOREIGN KEY (swh_prt_nr) REFERENCES swh_prts(swh_prt_id)
)type=InnoDB;

CREATE Table comp_hdds(
  comp_hdd_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_nr int(10),
  comp_hdd_size float(10),
  comp_hdd_left float(10),
  comp_hdd_mpoint varchar(30),
  comp_hdd_peak float(10),
  comp_hdd_peak_date int(11)#,
  # FOREIGN KEY (sys_nr) REFERENCES syss(sys_id)
)type=InnoDB;

CREATE Table comp_es(
  comp_e_id int(10) NOT NULL AUTO_INCREMENT PRIMARY KEY,
  sys_nr int(10),
  comp_e_temp int(10),
  comp_e_memusage int(10),
  comp_e_cpuusage int(10),
  comp_e_date int(11)#,
  # FOREIGN KEY (sys_nr) REFERENCES syss(sys_id)
)type=InnoDB;

# Insert different system-types into the table sys_types
INSERT INTO sys_types (sys_type_descr) VALUES(''Server'');
INSERT INTO sys_types (sys_type_descr) VALUES(''Workstation'');
INSERT INTO sys_types (sys_type_descr) VALUES(''Printer'');
INSERT INTO sys_types (sys_type_descr) VALUES(''Switch'');
INSERT INTO sys_types (sys_type_descr) VALUES(''Hub'');

# Insert different errors in prn_hrDeviceStatuss
INSERT INTO prn_hrDeviceStatuss (prn_hrDeviceStatus_id, prn_hrDeviceStatus_descr)
VALUES(1,``unknown'');
INSERT INTO prn_hrDeviceStatuss (prn_hrDeviceStatus_id, prn_hrDeviceStatus_descr)
VALUES(2,``running'');
INSERT INTO prn_hrDeviceStatuss (prn_hrDeviceStatus_id, prn_hrDeviceStatus_descr)
VALUES(3,``warning'');
INSERT INTO prn_hrDeviceStatuss (prn_hrDeviceStatus_id, prn_hrDeviceStatus_descr)
VALUES(4,``testing'');
INSERT INTO prn_hrDeviceStatuss (prn_hrDeviceStatus_id, prn_hrDeviceStatus_descr)
VALUES(5,``down'');

# Insert different errors in prn_hrPrinterStatuss
INSERT INTO prn_hrPrinterStatuss (prn_hrPrinterStatus_id, prn_hrPrinterStatus_descr)
```

```
VALUES(1,‘‘other’’);
INSERT INTO prn_hrPrinterStatuss (prn_hrPrinterStatus_id, prn_hrPrinterStatus_descr)
VALUES(2,‘‘unknown’’);
INSERT INTO prn_hrPrinterStatuss (prn_hrPrinterStatus_id, prn_hrPrinterStatus_descr)
VALUES(3,‘‘idle’’);
INSERT INTO prn_hrPrinterStatuss (prn_hrPrinterStatus_id, prn_hrPrinterStatus_descr)
VALUES(4,‘‘printing’’);
INSERT INTO prn_hrPrinterStatuss (prn_hrPrinterStatus_id, prn_hrPrinterStatus_descr)
VALUES(5,‘‘warmup’’);

# Insert different errors in prn_hrPrinterDetectedErrorStates
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(0,‘‘lowPaper’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(1,‘‘noPaper’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(2,‘‘lowToner’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(3,‘‘noToner’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(4,‘‘doorOpen’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(5,‘‘jammed’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(6,‘‘offline’’);
INSERT INTO prn_hrPrinterDetectedErrorStates (prn_hrPrinterDetectedErrorState_id,
prn_hrPrinterDetectedErrorState_descr) VALUES(7,‘‘serviceRequested’’);
```

D Skärmdumpar

Figure D.1: Översiktlig information om de övervakade noderna

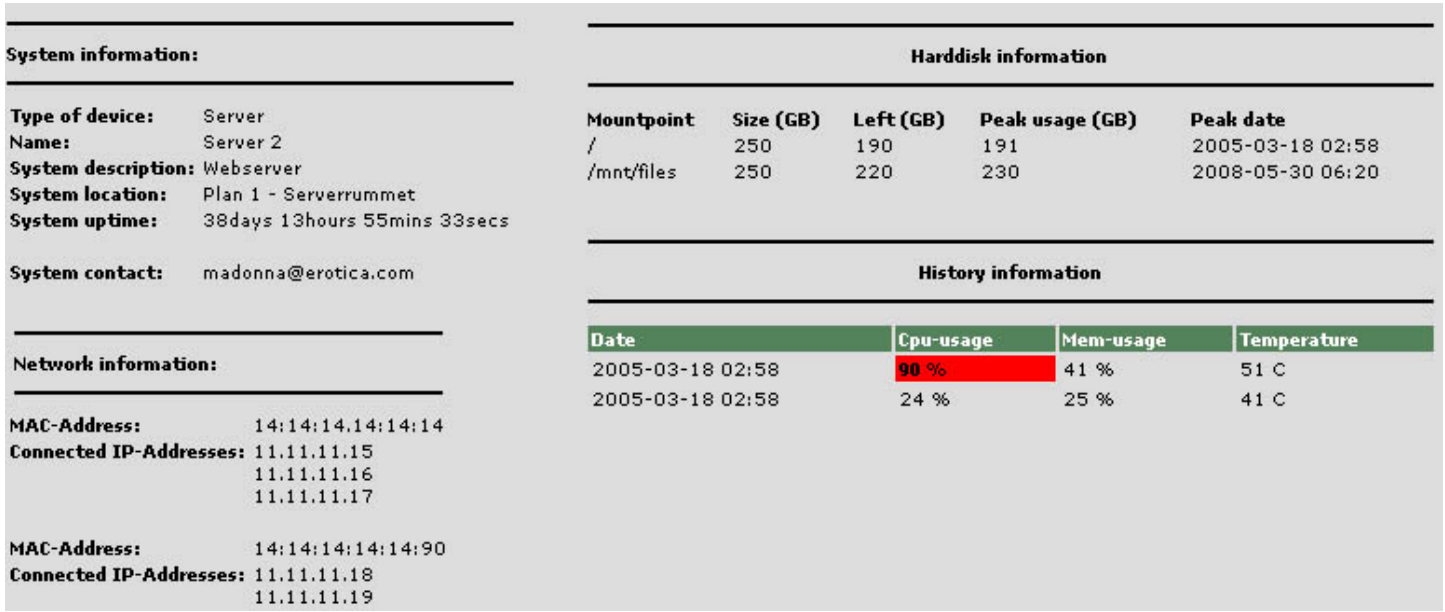

Figure D.2: Detaljerad information om den övervakad nod

E Skisser Gränssnitt

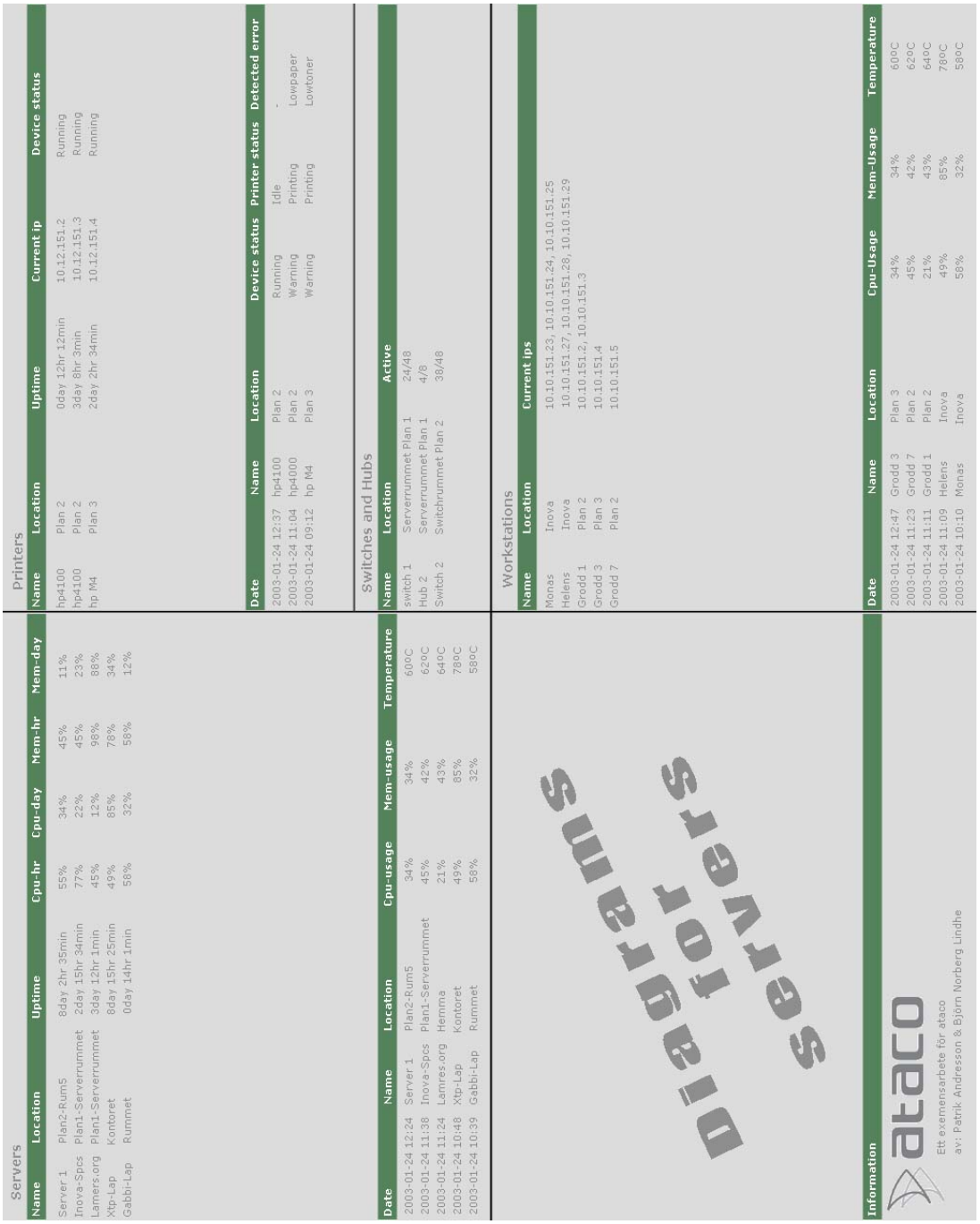

Figure E.1: Ursprunglig design av gränssnitt

F Förslag till insamling av SNMP-data

Nedan har de olika data som behöver samlas in och lagras från respektive typ av SNMP-agent sammanställts. I bilaga F.4 hittas en mer detaljerad beskrivning av de olika typerna av data som skall samlas in. De objekt som är markerade med (*) har ingen källa hittats, men är dock önskvärda att samla in.

F.1 Servrar & Arbetsstationer

1.3.6.1.2.1.1.1	sysDescr	Övergripande systembeskrivning
1.3.6.1.2.1.1.3	sysUpTime	Systemets uptime
1.3.6.1.2.1.1.4	sysContact	Kontaktperson för systemet
1.3.6.1.2.1.1.5	sysName	Den specifika nodens namn i SNMP-nätet
1.3.6.1.2.1.1.6	sysLocation	Den fysiska platsen där systemet befinner sig
?	atPhysAddress	Den fysiska adressen för NIC:et (MAC-adressen)
?	atNetAddress	Nätverksadressen för NIC:et (IP-adressen)
1.3.6.1.2.1.6.4	tcpMaxConn	Det maximala antalet TCP-connections som systemet har stöd för
1.3.6.1.2.1.6.5	tcpActiveOpens	Antalet TCP-uppkopplingar som har initierats aktivt
1.3.6.1.2.1.6.6	tcpPassiveOpens	Antalet TCP-uppkopplingar som har initierats passivt
1.3.6.1.2.1.6.10	tcpInSegs	Totala antalet segment som har tagits emot genom de för tillfället aktiva uppkopplingarna
1.3.6.1.2.1.6.11	tcpOutSegs	Totala antalet segment som har skickats genom de för tillfället aktiva uppkopplingarna
1.3.6.1.2.1.6.12	tcpRetransSegs	Det totala antalet segment som har behövt sändas om
1.3.6.1.2.1.6.14	tcpInErrs	Det totala antalet felaktiva segment som har

		tagits emot
1.3.6.1.2.1.7.2	udpNoPorts	Antalet udp-datagram som tagits emot där destinationsporten inte tillhandahållit någon serviceapplikation
1.3.6.1.2.1.7.1	udpInDatagrams	Antalet datagram som har levererats till UDP-användare
1.3.6.1.2.1.7.2	udpOutDatagrams	Antalet datagram som har skickars

Processorbelastning *
CPU klockfrekvens *
RAM-användande *
Hårddiskstatus *
Temperatur *
Fläktvarv *

F.2 Switchar och hubbar

1.3.6.1.2.1.1.1	sysDescr	Övergripande systebeskrivning
1.3.6.1.2.1.1.3	sysUpTime	Systemets uptime
1.3.6.1.2.1.1.4	sysContact	Kontaktperson för systemet
1.3.6.1.2.1.1.5	sysName	Den specifika nodens namn i SNMP-nätet
1.3.6.1.2.1.1.6	sysLocation	Den fysiska platsen där systemet befinner sig

Antalet aktiva portar *
Belastningen på de olika portarna *
Vilka hastigheter de olika portarna arbetar på (10/100 Mbit) *

F.3 Skrivare

1.3.6.1.2.1.1.1	sysDescr	Övergripande systembeskrivning
1.3.6.1.2.1.1.3	sysUpTime	Systemets uptime
1.3.6.1.2.1.1.4	sysContact	Kontaktperson för systemet
1.3.6.1.2.1.1.5	sysName	Den specifika nodens namn i SNMP-nätet

1.3.6.1.2.1.1.6	sysLocation	Den fysiska platsen där systemet befinner sig
?	atPhysAddress	Den fysiska adressen för NIC:et (MAC-adressen)
?	atNetAddress	Nätverksadressen för NIC:et (IP-adressen)
?	hrDeviceStatus	Enhetens operationella status
?	hrPrinterStatus	Printerstatusen för enheten
?	hrPrinterDetected ErrorState	Innehåller de felmeddelanden som uppstår

F.4 Källor

```
1.3.6.1.2.1.1.1 sysDescr
Källa:  http://www.alvestrand.no/objectid/1.3.6.1.2.1.1.1.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.1.1
OID description:
sysDescr OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION "A textual description of the entity.  This value should
    include the full name and version identification of the
    system's hardware type, software operating-system, and
    networking software.  It is mandatory that this only
    contain printable ASCII characters."
  ::= { system 1 }



1.3.6.1.2.1.1.3 - sysUpTime
http://www.alvestrand.no/objectid/1.3.6.1.2.1.1.3.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.1.3
OID description:
```

```
sysUpTime OBJECT-TYPE
  SYNTAX TimeTicks
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The time (in hundredths of a second) since the
    network management portion of the system was last
    re-initialized."
  ::= { system 3 }
```

```
1.3.6.1.2.1.1.4 - sysContact
http://www.alvestrand.no/objectid/1.3.6.1.2.1.1.4.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.1.4
OID description:
sysContact OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "The textual identification of the contact person
    for this managed node, together with information
    on how to contact this person."
  ::= { system 4 }
```

```
1.3.6.1.2.1.1.5 - sysName
http://www.alvestrand.no/objectid/1.3.6.1.2.1.1.5.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.1.5
```

```
OID description:
sysName OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "An administratively-assigned name for this
    managed node.  By convention, this is the node's
    fully-qualified domain name."
  ::= { system 5 }



1.3.6.1.2.1.1.6 - sysLocation
http://www.alvestrand.no/objectid/1.3.6.1.2.1.1.6.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.1.6
OID description:
sysLocation OBJECT-TYPE
  SYNTAX DisplayString (SIZE (0..255))
  ACCESS read-write
  STATUS mandatory
  DESCRIPTION
    "The physical location of this node (e.g.,
    `telephone closet, 3rd floor')."
  ::= { system 6 }



http://www.faqs.org/rfcs/rfc1213.html

3.3.  Physical Addresses
```

```
  As a further, textual convention in the MIB, the datatype

    PhysAddress ::=
      OCTET STRING

  is introduced to represent media- or physical-level addresses.

  The following objects are now defined in terms of PhysAddress:

    ifPhysAddress
    atPhysAddress
    ipNetToMediaPhysAddress

  It should be noted that this change has no effect on either the
  syntax nor semantics of these objects.  The use of the PhysAddress
  notation is merely an artifact of the explanatory method used in
  MIB-II and future MIBs.

-----

    atPhysAddress OBJECT-TYPE
      SYNTAX PhysAddress
      ACCESS read-write
      STATUS deprecated
      DESCRIPTION
        "The media-dependent 'physical' address.

        Setting this object to a null string (one of zero
        length) has the effect of invaliding the
        corresponding entry in the atTable object.  That
        is, it effectively dissasociates the interface
        identified with said entry from the mapping
        identified with said entry.  It is an
```

```
          implementation-specific matter as to whether the
          agent removes an invalidated entry from the table.
          Accordingly, management stations must be prepared
          to receive tabular information from agents that
          corresponds to entries not currently in use.
          Proper interpretation of such entries requires
          examination of the relevant atPhysAddress object."
        ::= { atEntry 2 }

---

    atNetAddress OBJECT-TYPE
      SYNTAX NetworkAddress
      ACCESS read-write
      STATUS deprecated
      DESCRIPTION
        "The NetworkAddress (e.g., the IP address)
        corresponding to the media-dependent 'physical'
        address."

      ::= { atEntry 3 }
```


http://www.ietf.org/rfc/rfc1759.txt

Printer Status	hrDeviceStatus	hrPrinterStatus	hrPrinterDetectedErrorState
Normal	running(2)	idle(3)	none set
Busy/ Temporarily Unavailable	running(2)	printing(4)	
Non Critical Alert Active	warning(3)	idle(3) or printing(4)	could be: lowPaper, lowToner, or serviceRequested
Critical Alert Active	down(5)	other(1)	could be: jammed, noPaper, noToner, coverOpen, or serviceRequested
Unavailable	down(5)	other(1)	
Moving off-line	warning(3)	idle(3) or printing(4)	offline
Off-line	down(5)	other(1)	offline
Moving on-line	down(5)	warmup(5)	
Standby	running(2)	other(1)	

2.2.13.2.1. Host MIB Printer Status

For completeness, the definitions of the Printer Status objects of the Host MIB are given below:

```
    hrDeviceStatus OBJECT-TYPE
      SYNTAX INTEGER {
        unknown(1),
        running(2),
        warning(3),
        testing(4),
        down(5)
}
      ACCESS read-only
      STATUS mandatory
      DESCRIPTION
        "The current operational state of the device
```



```
        described by this row of the table.  A value
        unknown(1) indicates that the current state of the
        device is unknown.  running(2) indicates that the
        device is up and running and that no unusual error
        conditions are known.  The warning(3) state
        indicates that agent has been informed of an
        unusual error condition by the operational software
        (e.g., a disk device driver) but that the device is
        still 'operational'.  An example would be high
        number of soft errors on a disk.  A value of
        testing(4), indicates that the device is not
        available for use because it is in the testing
```

```
      state.  The state of down(5) is used only when the
      agent has been informed that the device is not
      available for any use."
    ::= { hrDeviceEntry 5 }

  hrPrinterStatus OBJECT-TYPE
  SYNTAX INTEGER {
    other(1),
    unknown(2),
    idle(3),
    printing(4),
    warmup(5)
  }
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The current status of this printer device.  When
    in the idle(1), printing(2), or warmup(3) state,
    the corresponding hrDeviceStatus should be
    running(2) or warning(3).  When in the unknown
    state, the corresponding hrDeviceStatus should be
    unknown(1)."
  ::= { hrPrinterEntry 1 }

hrPrinterDetectedErrorState OBJECT-TYPE
  SYNTAX OCTET STRING
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "This object represents any error conditions
    detected by the printer.  The error conditions are
    encoded as bits in an octet string, with the
    following definitions:
```

Condition Bit # hrDeviceStatus

lowPaper 0 warning(3)
noPaper 1 down(5)
lowToner 2 warning(3)
noToner 3 down(5)
doorOpen 4 down(5)
jammed 5 down(5)
offline 6 down(5)
serviceRequested 7 warning(3)

If multiple conditions are currently detected and the hrDeviceStatus would not otherwise be unknown(1) or testing(4), the hrDeviceStatus shall correspond to the worst state of those indicated, where down(5) is worse than warning(3) which is worse than running(2).

Bits are numbered starting with the most significant bit of the first byte being bit 0, the least significant bit of the first byte being bit 7, the most significant bit of the second byte being bit 8, and so on. A one bit encodes that the condition was detected, while a zero bit encodes that the condition was not detected.

```
      This object is useful for alerting an operator to
      specific warning or error conditions that may
      occur, especially those requiring human
intervention."
    ::= { hrPrinterEntry 2 }
```

```
1.3.6.1.2.1.6.4 - tcpMaxConn
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.4.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.4
OID description:
tcpMaxConn OBJECT-TYPE
  SYNTAX INTEGER
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The limit on the total number of TCP connections
    the entity can support.  In entities where the
    maximum number of connections is dynamic, this
    object should contain the value -1."
  ::= { tcp 4 }
```

```
1.3.6.1.2.1.6.5 - tcpActiveOpens
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.5.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.5
OID description:
tcpActiveOpens OBJECT-TYPE
```

```
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of times TCP connections have made a
    direct transition to the SYN-SENT state from the
    CLOSED state."
  ::= { tcp 5 }
```

```
1.3.6.1.2.1.6.6 - tcpPassiveOpens
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.6.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.6
OID description:
tcpPassiveOpens OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The number of times TCP connections have made a
    direct transition to the SYN-RCVD state from the
    LISTEN state."
  ::= { tcp 6 }
```

```
1.3.6.1.2.1.6.10 - tcpInSegs
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.10.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.10
OID description:
```

```
tcpInSegs OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of segments received, including
    those received in error.  This count includes
    segments received on currently established
    connections."
  ::= { tcp 10 }
```

```
1.3.6.1.2.1.6.11 - tcpOutSegs
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.11.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.11
OID description:
tcpOutSegs OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of segments sent, including
    those on current connections but excluding those
    containing only retransmitted octets."
  ::= { tcp 11 }
```

```
1.3.6.1.2.1.6.12 - tcpRetransSegs
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.12.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
```

```
OID value:  1.3.6.1.2.1.6.12
OID description:
tcpRetransSegs OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of segments retransmitted - that
    is, the number of TCP segments transmitted
    containing one or more previously transmitted
    octets."
  ::= { tcp 12 }



1.3.6.1.2.1.6.14 - tcpInErrs
http://www.alvestrand.no/objectid/1.3.6.1.2.1.6.14.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.6.14
OID description:
tcpInErrs OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of segments received in error
    (e.g., bad TCP checksums)."
  ::= { tcp 14 }



1.3.6.1.2.1.7.2 - udpNoPorts
http://www.alvestrand.no/objectid/1.3.6.1.2.1.7.2.html
```

```
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.7.2
OID description:
udpNoPorts OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of received UDP datagrams for
    which there was no application at the destination
    port."
  ::= { udp 2 }
```

```
1.3.6.1.2.1.7.1 - udpInDatagrams
http://www.alvestrand.no/objectid/1.3.6.1.2.1.7.1.html
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.7.1
OID description:
udpInDatagrams OBJECT-TYPE
SYNTAX Counter
ACCESS read-only
STATUS mandatory
DESCRIPTION
"The total number of UDP datagrams delivered to
UDP users."
::= { udp 1 }
```

```
1.3.6.1.2.1.7.4 - udpOutDatagrams
http://www.alvestrand.no/objectid/1.3.6.1.2.1.7.4.html
```

```
Generated from OBJECT-TYPE definition found in rfc1213-mib2.asn1
OID value:  1.3.6.1.2.1.7.4
OID description:
udpOutDatagrams OBJECT-TYPE
  SYNTAX Counter
  ACCESS read-only
  STATUS mandatory
  DESCRIPTION
    "The total number of UDP datagrams sent from this
    entity."
  ::= { udp 4 }
```

G Kod

/index.php

```
<?
  session_start();
  include(‘‘include/alarms.php’’);
?>
<html>
  <head>
    <title>Exjobb</title>
    <link rel=‘‘styleshee’’ href=‘‘include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <frameset cols=‘‘*,1200,*’’ frameborder=0>
    <frame src=‘‘tom.php’’ scrolling=no noresize></frame>
    <frameset rows=‘‘*,800,*’’ bordercolor=‘‘black’’ bordersize=‘‘2’’>
        <frame src=‘‘tom.php’’ scrolling=no noresize></frame>
        <frameset cols=‘‘50%,50%’’ frameborder=1>
        <frameset rows=‘‘300,100,300,100’’>
            <frame src=‘‘srv/main.php’’ name=‘‘srv_main’’ noresize></frame>
            <frame src=‘‘srv/events.php’’ name=‘‘srv_events’’ noresize></frame>
            <frame src=‘‘srv/diagrams.php’’ name=‘‘srv_diagrams’’ noresize></frame>
            <frame src=‘‘info.php’’ name=‘‘info’’ noresize></frame>
        </frameset>
        <frameset rows=‘‘200,100,100,300,100’’>
          <frame src=‘‘printer/main.php’’ name=‘‘printer_main’’ noresize></frame>
          <frame src=‘‘printer/events.php’’ name=‘‘printer_events’’ noresize></frame>
          <frame src=‘‘swh/main.php’’ name=‘‘swh_main’’ noresize></frame>
          <frame src=‘‘wrk/main.php’’ name=‘‘wrk_main’’ noresize></frame>
          <frame src=‘‘wrk/events.php’’ name=‘‘wrk_events’’ noresize></frame>
        </frameset>
      </frameset>
      <frame src=‘‘tom.php’’ scrolling=no noresize></frame>
    </frameset>
    <frame src=‘‘tom.php’’ scrolling=no noresize></frame>
```

```
  </frameset>
</html>
```

/tom.php

```
<?
  session_start();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>tom sida</title>
  </head>
  <body bgcolor=#DCDCDC>

  </body>
</html>
```


/info.php

```
<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Information</title>
    <link rel=‘‘stylesheet’’ href=‘‘include/style
main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100\’’ height=‘‘100\’’ border=‘‘0’’>
      <tr width=‘‘50\’’ height=‘‘100\’’ align=‘‘left’’ valign=‘‘middle’’>
        <td>
          <img src=‘‘logo.gif’’>
```

```
        </td>
        <td width=‘‘50\’’ height=‘‘100\’’ align=‘‘left’’ valign=‘‘middle’’>
          <font class=‘‘text-bold’’>
            Ett examensarbete vid:  <br>
          </font>
          <a href=‘‘http://www.kau.se’’>Karlstads Universitet</a> <br>
          Hösten 2003 <br>
          <font class=‘‘text-bold’’>
            Av:
          </font>
          <a href=‘‘mailto:patrik@ataco.se’’>Patrik Andersson</a> och
          <a href=‘‘mailto:bjorn@brunnen.se’’>Björn N. Lindhe</a>
        </td>
      </tr>
    </table>
  </body>
</html>
```


/include/alarms.php

```
<?
  //This file contains the trigger-levels for the different alarms

  //CPU-Temperature
  $_SESSION[‘‘cpu_usage’’] = 90;

  //Main-memory usage
  $_SESSION[‘‘mem_usage’’] = 80;

  //CPU-Temperature
  $_SESSION[‘‘cpu_temp’’] = 55;

  //Hdd-usage
```

```
  $_SESSION[‘‘hdd_left’’] = 10;

  //Active ports on switches and hubs
  $_SESSION[‘‘port_usage’’] = 10;
?>
```

/include/class.php

```
<?
  include(‘‘databasen.php’’);
  include(‘‘functions.php’’);

  //Superclass - This class is inherited by all the other classes
  class superClass
  {
    var $rs, $result;

    function superClass($sql)
    {
      $this->result = mysql_query($sql)
        or die(mysql_error());
    }

    function getnext ()
    {
      if($this->result)
        $this->rs = mysql_fetch_array($this->result);
      return $this->rs;
    }

    function seek ($position)
    {
      if($this->result)
```

```
      mysql_data_seek($this->result, $position);
  }

  function setzero ()
  {
    if($this->result)
      mysql_data_seek($this->result, 0);
  }
}

//Class for selecting specific informtion about a system
class specific_information extends superClass
{
  function specific_information ($sys_id)
  {
    $this->superClass(‘‘SELECT *, sys_types.sys_type_descr
      FROM syss
      INNER JOIN sys_types ON syss.sys_type_nr = sys_types.sys_type_id
      WHERE sys_id = ‘‘.$sys_id.’’;’’);
  }
}

//Class for selecting all NICs for a system
class nics extends superClass
{
  function nics ($sys_id)
  {
    $this->superClass(‘‘SELECT *
      FROM sys_nics
      WHERE sys_nr = ‘‘.$sys_id.’’;’’);
  }
}
```

```
//Class for selecting all IP-Addresses for a NIC
class ips extends superClass
{
  function ips ($sys_nic_id)
  {
    $this->superClass(‘‘SELECT *
      FROM sys_ips
      WHERE sys_nic_nr = ‘‘.$sys_nic_id.’’;’’);
  }
}

//Class for selecting information about hdds connected to a workstation or a server
class hdds extends superClass
{
  function hdds ($sys_id)
  {
    $this->superClass(‘‘SELECT *
      FROM comp_hdds
      WHERE sys_nr = ‘‘.$sys_id.’’;’’);
  }
}

//Class for selecting all events about a specific workstation or server
class comp_events extends superClass
{
  function comp_events ($sys_id)
  {
    $this->superClass(‘‘SELECT *
      FROM comp_es
      WHERE sys_nr = ‘‘.$sys_id.’’
      ORDER BY comp_e_date DESC;’’);
  }
}
```

```
?>
```

/include/databasen.php

```
<?
  $status = mysql_connect(‘‘194.23.220.100’’, ‘‘snmpdata’’, ‘‘davc19’’);
  if (!$status)
  die(‘‘Error opening mySQL-connection’’);

  $database = ‘‘snmpdata’’;

  mysql_select_db($database, $status)
    or die(‘‘Error opening databse:  ‘‘.$database);
?>
```


/include/functions.php

```
<?
  //Function for printing network information about a system
  function network_information($sys_id)
  {
    $nics = new nics($sys_id);
    print ‘‘<table cellspacing=\‘‘0\’’ cellpadding=\‘‘0\’’ border=\‘‘0\’’>
      <tr>
        <td colspan=\‘‘2\’’>
          <table width=\‘‘100%\’’ cellspacing=\‘‘0\’’ cellpadding=\‘‘0\’’ border=\‘‘0\’’>
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td>
                <font class=\‘‘text-bold\’’>
                  Network information:
```

```
            </font>
          </td>
        </tr>
        <tr>
          <td><hr></td>
        </tr>
      </table>
    </td>
  </tr>’’;

while($rs = $nics->getnext())
{
  print ‘‘<tr>’’;
  print ‘‘<td>’’;
  print ‘‘<font class=\‘‘text-bold\’’>’’;
  print ‘‘MAC-Address:’’;
  print ‘‘</font>’’;

  print ‘‘</td>’’;
  print ‘‘<td>’’;

  print $rs[‘‘sys_nic_PhysAddress’’];

  print ‘‘</td>’’;
  print ‘‘</tr>’’;
  print ‘‘<tr>’’;
  print ‘‘<td valign=\‘‘top\’’>’’;

  print ‘‘<font class=\‘‘text-bold\’’>’’;
  print ‘‘Connected IP-Addresses:’’;
  print ‘‘</font>’’;

  print ‘‘</td>’’;
```

```
    print ‘‘<td>’’;

    $nic_ips = new ips($rs[‘‘sys_nic_id’’]);

    while($rs2 = $nic_ips->getnext())
      print $rs2[‘‘sys_ip_NetAddress’’].’’<br>’’;

    print ‘‘</td>’’;
    print ‘‘</tr>’’;
    print ‘‘<tr>’’;
    print ‘‘<td>’’;

    print ‘‘ ’’;

    print ‘‘</td>’’;
    print ‘‘<td>’’;

    print ‘‘ ’’;

    print ‘‘</td>’’;
    print ‘‘<td>’’;
    print ‘‘</tr>’’;
  }
  print ‘‘</table>’’;
}

//Function for calculating uptime in days, hours, minutes, seconds.
function uptime($uptime)
{
  $days = $uptime / 86400;
  settype($days, ‘‘int’’);

  $sec_remaning = $uptime % 86400;
```

```
$hours = $sec_remaning / 3600;
settype($hours, ‘‘int’’);

$sec_remaning = $uptime % 3600;

$minutes = $sec_remaning / 60;
settype($minutes, ‘‘int’’);

$seconds = $sec_remaning % 60;

$new_uptime = $days;

if($days > 1)
  $new_uptime .= ‘‘days ‘‘;
else
  $new_uptime .= ‘‘day ‘‘;

$new_uptime .= $hours;

if($hours > 1)
  $new_uptime .= ‘‘hours ‘‘;
else
  $new_uptime .= ‘‘hour ‘‘;

$new_uptime .= $minutes;

if($minutes > 1)
  $new_uptime .= ‘‘mins ‘‘;
else
  $new_uptime .= ‘‘min ‘‘;

$new_uptime .= $seconds;
```

```
    if($seconds > 1)
      $new_uptime .= ‘‘secs ‘‘;
    else
      $new_uptime .= ‘‘sec ‘‘;

    return $new_uptime;
  } ?>
```

/include/style_main.css

```
a
{
  font-size :  9px;
  font-family :  Verdana, Geneva, Arial, Helvetica, sans-serif;
  background :  White;
  background-color :  #DCDCDC;
  text-decoration :  none;
  color :  Black;
}

a:hover
{
text-decoration :  underline;
}

table
{
  border :  none;
}

td
{
```

```
  background :  #DCDCDC;
  font-family :  Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-size :  9px;
  padding :  1 2 1 2;
}

.text-bold
{
font-family :  Verdana, Geneva, Arial, Helvetica, sans-serif;
font-size :  9px;
font-weight :  bold;
}

th
{
  background :  #54835B;
  font-family :  Verdana, Geneva, Arial, Helvetica, sans-serif;
  font-size :  9px;
  font-weight :  bold;
  color :  white;
  border :  thin;
  text-align :  left;
  padding :  1 2 1 2;
}

body
{
  margin :  2 2 2 2;
  font-size :  12px;
  background-color :  #DCDCDC;
}

hr
```

```
{
color :  Black;
height :  2 px;
}

.frameborder
{
color :  black;
width :  2 px;
}
```

/printer/class.php

```
<?
  include(‘‘../include/class.php’’);

  //Class for selecting main-information about printers
  class printers extends superClass
  {
    function printers ()
    {
      $this->superClass(‘‘
        SELECT *
        FROM syss
        INNER JOIN sys_types
          ON syss.sys_type_nr = sys_types.sys_type_id
        WHERE sys_type_id = 3;’’);
    }
  }

  class printer_specific extends superClass
  {
    function printer_specific ($sys_id)
```

```
  {
    $this->superClass(‘‘
      SELECT *
      FROM syss
      INNER JOIN sys_types
        ON syss.sys_type_nr = sys_types.sys_type_id
      WHERE sys_id = ‘‘.$sys_id.’’;’’);
  }
}

class printer_events extends superClass
{
  function printer_events ($sys_id)
  {
    $this->superClass(‘‘
      SELECT *
      FROM prn_es as P
      LEFT JOIN syss AS S
        ON P.sys_nr=S.sys_id
      LEFT JOIN prn_hrDeviceStatuss AS DS
        ON P.prn_e_hrDeviceStatus_nr=DS.prn_hrDeviceStatus_id
      LEFT JOIN prn_hrPrinterStatuss AS PS
        ON P.prn_e_hrPrinterStatus_nr=PS.prn_hrPrinterStatus_id
      LEFT JOIN prn_hrPrinterDetectedErrorStates AS DES
        ON P.prn_e_hrPrinterDetectedErrorState_nr=DES.prn_hrPrinterDetectedErrorState_id
      WHERE sys_nr = ‘‘.$sys_id.’’
      ORDER BY prn_e_date DESC;’’);
  }
}

class printer_events_last extends superClass
{
  function printer_events_last( $num_of_entries )
```

```
    {
      $this->superClass(‘‘
        SELECT *
        FROM prn_es as P
        LEFT JOIN syss AS S
          ON P.sys_nr=S.sys_id
        LEFT JOIN prn_hrDeviceStatuss AS DS
          ON P.prn_e_hrDeviceStatus_nr=DS.prn_hrDeviceStatus_id
        LEFT JOIN prn_hrPrinterStatuss AS PS
          ON P.prn_e_hrPrinterStatus_nr=PS.prn_hrPrinterStatus_id
        LEFT JOIN prn_hrPrinterDetectedErrorStates AS DES
          ON P.prn_e_hrPrinterDetectedErrorState_nr=DES.prn_hrPrinterDetectedErrorState_id
        ORDER BY prn_e_date DESC
        LIMIT $num_of_entries’’);
    }
  }
?>
```


/printer/events.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $printer_events_last = new printer_events_last(4);
?>

<html>
  <head>
    <link href=../include/style_main.css rel=stylesheet type=text/css>
  </head>
  <body>
    <table border=0 width=100%>
      <tr>
```

```
        <th>Date</th>
        <th>Name</th>
        <th>Device status</th>
        <th>Printer status</th>
        <th>Detected error</th>
      </tr>

<?php
  while( $rs = $printer_events_last->getnext() )
  {
    print ‘‘<tr>’’;
    print ’’ <td>’’.date(‘‘Y-m-d H:i’’, $rs[‘‘prn_e_date’’]).’’</td>’’;
    print ’’ <td>’’.$rs[‘‘sys_sysName’’].’’</td>’’;
    print ’’ <td ‘‘;

    if($rs[‘‘prn_e_hrDeviceStatus_nr’’] == 5)
      print ‘‘style=\‘‘background-color :  red;
font-weight :  bold\‘‘’’;

    if($rs[‘‘prn_e_hrDeviceStatus_nr’’] == 1 ||
      $rs[‘‘prn_e_hrDeviceStatus_nr’’] == 3)
      print ‘‘style=\‘‘background-color :  yellow;
font-weight :  bold\‘‘’’;

    print ’’>’’.$rs[‘‘prn_hrDeviceStatus_descr’’].’’</td>’’;
    print ’’ <td ‘‘;

    if($rs[‘‘prn_e_hrPrinterStatus_nr’’] == 2)
      print ‘‘style=\‘‘background-color :  yellow;
font-weight :  bold\‘‘’’;

    print ’’>’’.$rs[‘‘prn_hrPrinterStatus_descr’’].’’</td>’’;
    print ’’ <td ‘‘;
```

```
    if($rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 1
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 3
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 4
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 5
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 6)
        print ‘‘style=\‘‘background-color :  red;
font-weight :  bold\‘‘’’;

    if($rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 0
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 2
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 7)
        print ‘‘style=\‘‘background-color :  yellow;
font-weight :  bold\‘‘’’;

    print ’’>’’.$rs[‘‘prn_hrPrinterDetectedErrorState_descr’’].’’</td>’’;
    print ‘‘</tr>’’;
    }
?>
    </table>
  </body>
</html>
```

/printer/main.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $printers = new printers();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
```

```
  <head>
    <link href=../include/style_main.css rel=stylesheet type=text/css>

    <script language=‘‘JavaScript’’>
      function open_printer_info()
      {
        window.open(‘‘../tom.php’’, ‘‘printer_info_window’’,
‘‘fullscreen=no,toolbar=no,status=yes,menubar=no,scrollbars=no,resizable=no,directories=no,location=no,width=780,height=560,left=
        document.printer_info_send.submit();
      }

    </script>
  </head>
  <body>

  <font class=‘‘text-bold’’>Printers</font>
  <table border=0 width=100%>
    <tr>
      <th>Name</th>
      <th>Location</th>
      <th>Uptime</th>
      <th>Device status</th>
      <th>Status date</th>
    </tr>

<?
  while($rs = $printers->getnext())
  {
    print ’’ <tr>’’;
    print ’’ <td>’’;
    print ’’ <a href=\‘‘#\’’
onclick=\‘‘document.printer_info_send.sys_id.value=‘‘.$rs[‘‘sys_id’’].’’;open_printer_info();\’’>’’.$rs[‘‘sys_sysName’’].’’</a>’’;
    print ’’ </td>’’;
```

```
    print '' <td>''.$rs[''sys_sysLocation''].''</td>'';

    print ''<td>''.  uptime($rs[''sys_sysUpTime'']) .''</td>'';

    $printer_event = new printer_events($rs[''sys_id'']);
    $rs2 = $printer_event->getnext();

    print ''<td '';
    if($rs2[''prn_e_hrDeviceStatus_nr''] == 5)
      print ''style=\''background-color :  red;
font-weight :  bold\'''';

    if($rs2[''prn_e_hrDeviceStatus_nr''] == 1 || $rs2[''prn_e_hrDeviceStatus_nr''] == 3)
      print ''style=\''background-color :  yellow;
font-weight :  bold\'''';

    print ''>'';
    print $rs2[''prn_hrDeviceStatus_descr''];
    print ''</td>'';

    print ''<td>'';
    print date(''Y-m-d H:i'', $rs2[''prn_e_date'']);
    print ''</td>'';
    print ''</tr>'';
  }
?>

     </table>
    <form name=''printer_info_send'' action=''printer_info.php'' target=''printer_info_window'' method=''POST''>
      <input type=''hidden'' name=''sys_id'' value=''''>
    </form>
  </body>
</html>
```

/printer/printer_info.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $sys_id = addslashes($_POST[‘‘sys_id’’]);
  $printer_information = new printer_specific($sys_id);
  $rs = $printer_information->getnext();
  $nics = new nics($sys_id);
  $printer_events = new printer_events($sys_id);
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Specific information</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘1’’>
      <tr>
        <td width=‘‘40%’’ align=‘‘left’’ valign=‘‘top’’>
          <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td><font class=‘‘text-bold’’>System information:</font></td>
            </tr>
            <tr>
              <td><hr></td>
            </tr>
```

```
            <tr>
              <td>
                <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                  <tr>
                    <td><font class=‘‘text-bold’’>Type of device:</font></td>
                    <td>
<?
  print $rs[‘‘sys_type_descr’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>Name:</font></td>
                    <td>
<?    print $rs[‘‘sys_sysName’’]; ?>                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System description:</font></td>
                    <td>
<?
  print $rs[‘‘sys_sysDescr’’];
?>                      </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System location:</font></td>
                    <td>
<?
  print $rs[‘‘sys_sysLocation’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System uptime:</font></td>
```

```
                    <td>
<?    print uptime($rs[‘‘sys_sysUpTime’’]); ?>                    </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td> </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System contact:</font></td>
                    <td>
<?
  print ‘‘<a href=\‘‘mailto:’’.$rs[‘‘sys_sysContact’’].’’\’’>’’.$rs[‘‘sys_sysContact’’].’’</a>’’;
?>                      </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td> </td>
                  </tr>
                </table>
              </td>
            </tr>
            <tr>
              <td>
                <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                  <tr>
                    <td colspan=‘‘2’’>
                      <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                        <tr>
                          <td><hr></td>
                        </tr>
                        <tr>
                          <td><font class=‘‘text-bold’’>Network information:</font></td>
                        </tr>
```

```
                        <tr>
                          <td><hr></td>
                        </tr>
                      </table>
                    </td>
                  </tr>
<?
  while($rs = $nics->getnext())
  {
    print '' <tr>'';
    print '' <td>'';
    print '' <font class=\''text-bold\''>'';
    print '' MAC-Address:'';
    print '' </font>'';
    print '' </td>'';
    print '' <td>'';
    print $rs['sys_nic_PhysAddress''];
    print '' </td>'';
    print ''</tr>'';
    print ''<tr>'';
    print ''<td valign=\''top\''>'';
    print ''<font class=\''text-bold\''>'';
    print ''Connected IP-Addresses:'';
    print ''</font>'';
    print ''</td>'';
    print ''<td>'';
    $nic_ips = new ips($rs['sys_nic_id'']);
    while($rs2 = $nic_ips->getnext())
      print $rs2['sys_ip_NetAddress''].''<br>'';
    print ''</td>'';
    print ''</tr>'';
    print ''<tr>'';
    print ''<td>'';
```

```
        print ‘‘ ’’;
        print ‘‘</td>’’;
        print ‘‘<td>’’;
        print ‘‘ ’’;
        print ‘‘</td>’’;
        print ‘‘<td>’’;
      }
    ?>
                    </table>
                  </td>
                </tr>
              </table>
            </td>
            <td width=‘‘60%’’ align=‘‘center’’ valign=‘‘top’’>
              <table border=0 width=100%>
                <tr>
                  <td colspan=‘‘5’’></td><hr></td>
                </tr>
                <tr>
                  <td align=‘‘center’’ colspan=‘‘5’’>
                    <font class=‘‘text-bold’’>History information</font>
                  </td>
                </tr>
                <tr>
                  <td colspan=‘‘5’’><hr></td>
                </tr>
                <tr>
                  <th>Date</th>
                  <th>Name</th>
                  <th>Device status</th>
                  <th>Printer status</th>
                  <th>Detected error</th>
                </tr>
```

```
<?php
  while( $rs = $printer_events->getnext() )
  {
    print ‘‘<tr>’’;
    print ’’ <td>’’.date(‘‘Y-m-d H:i’’, $rs[‘‘prn_e_date’’]).’’</td>’’;
    print ’’ <td>’’.$rs[‘‘sys_sysName’’].’’</td>’’;
    print ’’ <td ‘‘;
    if($rs[‘‘prn_e_hrDeviceStatus_nr’’] == 5)
      print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
    if($rs[‘‘prn_e_hrDeviceStatus_nr’’] == 1
      || $rs[‘‘prn_e_hrDeviceStatus_nr’’] == 3)
    print ‘‘style=\‘‘background-color :  yellow; font-weight :  bold\’’’;
    print ’’>’’.$rs[‘‘prn_hrDeviceStatus_descr’’].’’</td>’’;
    print ’’ <td ‘‘;
    if($rs[‘‘prn_e_hrPrinterStatus_nr’’] == 2)
      print ‘‘style=\‘‘background-color :  yellow; font-weight :  bold\’’’;
    print ’’>’’.$rs[‘‘prn_hrPrinterStatus_descr’’].’’</td>’’;
    print ’’ <td ‘‘;
    if($rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 1
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 3
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 4
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 5
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 6)
        print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
    if($rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 0
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 2
      || $rs[‘‘prn_e_hrPrinterDetectedErrorState_nr’’] == 7)
        print ‘‘style=\‘‘background-color :  yellow; font-weight :  bold\’’’;
    print ’’>’’.$rs[‘‘prn_hrPrinterDetectedErrorState_descr’’].’’</td>’’;
    print ‘‘</tr>’’;
  }
?>
          </table>
```

```
        </td>
      </tr>
    </table>
  </body>
</html>
```

/srv/class.php

```
<?
  include(‘‘../include/class.php’’);

  //Class for selecting main-information about servers
  class srvs extends superClass
  {
    function srvs ()
    {
      $this->superClass(‘‘SELECT *
        FROM syss
        INNER JOIN sys_types
          ON syss.sys_type_nr = sys_types.sys_type_id
        WHERE sys_type_id = 1;’’);
    }
  }

  //Class for selecting the last four events for all the servers
  class srvs_events extends superClass
  {
    function srvs_events ()
    {
      $this->superClass(‘‘SELECT *
        FROM comp_es
        INNER JOIN syss
          ON syss.sys_id = comp_es.sys_nr
```

```
        INNER JOIN sys_types
          ON sys_types.sys_type_id = syss.sys_type_nr
        WHERE sys_types.sys_type_id = 1
        ORDER BY comp_es.comp_e_date DESC
        LIMIT 4;'');
    }
  }

  //Class for selecting the harddisk-status
  class srvs_hddstatus extends superClass
  {
    function srvs_hddstatus ()
    {
      $this->superClass(``
        SELECT * FROM comp_hdds
        INNER JOIN syss
          ON syss.sys_id = comp_hdds.sys_nr
        INNER JOIN sys_types
          ON sys_types.sys_type_id = syss.sys_type_nr
        WHERE sys_types.sys_type_id = 1'');
    }
  }
?>
```

/srv/diagrams.php

```
<?php
  // Start session
  session_start();

  include(``class.php'');
  // Construct the class variabel
  $srvs_hddstatus = new srvs_hddstatus();
```

```
// Specify image parameters
$GraphWidth = 480;
$GraphHeight = 200;
$GraphScale = 2;
$GraphFont = 2;
// Insert data from database into an array for use when deciding the size
// of the bars
$GraphData = array();
for($i=0 ; $rs = $srvs_hddstatus->getnext() ; $i++)
{
  // Server name
  $GraphData[$i][server] = $rs[sys_sysName];
  // Where the harddisk is mounted
  $GraphData[$i][mpoint] = $rs[comp_hdd_mpoint];
  // How much space is left using perecent
  $GraphData[$i][space] =
    100* ($rs[comp_hdd_size] - $rs[comp_hdd_left]) / $rs[comp_hdd_size];
}

// Create the image structure
$image = imagecreate($GraphWidth, $GraphHeight);

// Define colors of different parts of the image
$colorBody = imagecolorallocate($image, 0xDC, 0xDC, 0xDC); //Gray
$colorGrid = imagecolorallocate($image, 0x66, 0x66, 0x66); //Dark grey
$colorBar = imagecolorallocate($image, 0x54, 0x83, 0x5B); //Green
$colorAlarmBar = imagecolorallocate($image, 0xED, 0x43, 0x43); //Red
$colorText = imagecolorallocate($image, 0x00, 0x00, 0x00); //Svart
$colorBarText = imagecolorallocate($image, 0xFF, 0xFF, 0xFF); //White

// Fill background of image with the color specified above
imagefill($image, 0, 0, $colorBody);
```

```
// Define size of fonts used to print insiide the diagram
$GridLabelWidth = imagefontwidth($GraphFont)*3+1;
//Vertical line between row-text and bars
imageline($image,
  $GridLabelWidth, 0,
  $GridLabelWidth, $GraphHeight-1,
  $colorGrid);

// Define color of the grid
$styleDashed = array_merge(array_fill(0, 4, $colorGrid),
  array_fill(0, 4, IMG_COLOR_TRANSPARENT));
imagesetstyle($image, $styleDashed);
//Horizontal line below bars
for($index = 0 ; $index < $GraphHeight; $index += $GraphHeight/10)
{
  imageline($image, 0, $index, $GraphWidth-1, $index, IMG_COLOR_STYLED);
  imagestring($image, $GraphFont, 0, $index,
    round(($GraphHeight - $index)/$GraphScale), $colorText);
}

// Horizontal line below bars
imageline($image, 0, $GraphHeight-1, $GraphWidth-1, $GraphHeight-1, $colorGrid);
// Define width of the bars
$BarWidth = (($GraphWidth-$GridLabelWidth)/count($GraphData)) - 10;

// For each bar;
for( $i=0 ; $i<count($GraphData) ; $i++ )
{
  $label1 = $GraphData[$i][server];
  $label2 = $GraphData[$i][mpoint];
  $value = $GraphData[$i][space];
  $BarTopX = $GridLabelWidth + (($i+1) * 10) + ($i * $BarWidth);
```

```
    $BarBottomX = $BarTopX + $BarWidth;
    $BarBottomY = $GraphHeight-1;
    $BarTopY = $BarBottomY - ($value * $GraphScale);
    //Fill the bars
    if($value<100-$_SESSION[hdd_left])
      imagefilledrectangle($image, $BarTopX, $BarTopY, $BarBottomX, $BarBottomY, $colorBar);
    else
      imagefilledrectangle($image, $BarTopX, $BarTopY, $BarBottomX, $BarBottomY, $colorAlarmBar);

    $LabelX = $BarTopX + (($BarBottomX - $BarTopX)/2) - (imagefontheight($GraphFont)/2);
    $LabelY = $BarBottomY-10;

    // Print the par itself
    imagestringup($image, $GraphFont, $LabelX, $LabelY, ‘‘$label1, $label2:  $value%’’, $colorBarText);
  }

  // Create the diagram image using the above code
  imagepng($image, ‘‘diagram.png’’);
  // Remove the image data from memory
  imagedestroy($image);
?>

<html>
<head>
  <title>Server diagrams</title>
  <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
</head>

<body>
<table width=‘‘100%’’ height=‘‘100%’’>
  <tr>
    <td width=‘‘100%’’ height=‘‘100%’’ align=‘‘center’’ valign=‘‘middle’’>
```

```
    <img src=diagram.png>

    </td>
  </tr>
</table>
</body>
</html>
```

/srv/events.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $srvs_events = new srvs_events();
?>
<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Workstations - Events</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’>
      <tr>
        <th>Date</th>
        <th>Name</th>
        <th>Location</th>
        <th>Cpu-usage</th>
        <th>Mem-usage</th>
        <th>Temperature</th>
      </tr>
<?
while($rs = $srvs_events->getnext())
```

```
{
  print ‘‘
    <tr>
      <td>
        <font class=\‘‘font-normal\’’>’’;
          $date_event = date(‘‘Y-m-d H:i’’,$rs[‘‘comp_e_date’’]);
          print $date_event.’’
        </font>
      </td>
      <td>
        <font class=\‘‘font-normal\’’>’’.$rs[‘‘sys_sysName’’].’’</font>
      </td>
      <td>
        <font class=\‘‘font-normal\’’>’’.$rs[‘‘sys_sysLocation’’].’’</font>
      </td>
      <td ‘‘;
  if($rs[‘‘comp_e_cpuusage’’] >= $_SESSION[‘‘cpu_usage’’])
    print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
  print  ‘‘  >
        <font class=\‘‘font-normal\’’>’’.$rs[‘‘comp_e_cpuusage’’].’’  %</font>
      </td>
      <td ‘‘;
  if($rs[‘‘comp_e_memusage’’] >= $_SESSION[‘‘mem_usage’’])
    print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
  print  ‘‘  >
        <font class=\‘‘font-normal\’’>’’.$rs[‘‘comp_e_memusage’’].’’  %</font>
      </td>
      <td ‘‘;
  if($rs[‘‘comp_e_temp’’] >= $_SESSION[‘‘cpu_temp’’])
    print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
  print  ‘‘  >
        <font class=\‘‘font-normal\’’>’’.$rs[‘‘comp_e_temp’’].’’  C</font>
      </td>
```

```
    </tr>'';
  }
?>
    </table>
  </body>
</html>
```

/srv/main.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $srvs = new srvs();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Workstations</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
    <script language=‘‘JavaScript’’>
      function open_srv_info()
      {
        window.open(‘‘../tom.php’’, ‘‘srv_info_window’’,
‘‘fullscreen=no,toolbar=no,status=yes,menubar=no,scrollbars=no,resizable=no,directories=no,location=no,width=780,height=560,left=
        document.srv_info_send.submit();
      }     </script>
  </head>
  <body>
    <font class=‘‘text-bold’’>Servers</font>
    <table width=‘‘100%’’>
      <tr>
        <th>Name</th>
```

```
        <th>Location</th>
        <th>Current IP-Addresses</th>
      </tr>
<?
  while($rs = $srvs->getnext())
  {
    $nics = new nics($rs[‘‘sys_id’’]);

    print ‘‘<tr>’’;

    print ‘‘<td>’’;
    print ‘‘<a href=\‘‘#\’’
onclick=\‘‘document.srv_info_send.sys_id.value=‘‘.$rs[‘‘sys_id’’].’’;open_srv_info();\’’>’’.$rs[‘‘sys_sysName’’].’’</a>’’;
    print ‘‘</td>’’;

    print ‘‘<td>’’;
    print $rs[‘‘sys_sysLocation’’];
    print ‘‘</td>’’;

    print ‘‘<td>’’;
    while($rs2 = $nics->getnext())
    {
      $ips = new ips($rs2[‘‘sys_nic_id’’]);
      while($rs3 = $ips->getnext())
      {
        print $rs3[‘‘sys_ip_NetAddress’’].’’, ‘‘;
      }
    }
    print ‘‘</td>’’;

    print ‘‘</tr>’’;
  }
?>
```

```
    </table>
    <form name=‘‘srv_info_send’’ action=‘‘srv_info.php’’ target=‘‘srv_info_window’’ method=‘‘POST’’>
      <input type=‘‘hidden’’ name=‘‘sys_id’’ value=‘‘’’>
    </form>
  </body>
</html>
```

/srv/srv_info.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $sys_id = addslashes($_POST[‘‘sys_id’’]);
  $information = new specific_information($sys_id);
  $rs = $information->getnext();
  $hdds = new hdds($sys_id);
  $srv_events = new comp_events($sys_id);
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Specific information</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘1’’>
      <tr>
        <td width=‘‘40%’’ align=‘‘left’’ valign=‘‘top’’>
          <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
            <tr>
              <td><hr></td>
            </tr>
```

```
            <tr>
              <td><font class=‘‘text-bold’’>System information:</font></td>
            </tr>
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td>
                <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                  <tr>
                    <td><font class=‘‘text-bold’’>Type of device:</font></td>
                    <td>
<?
  print $rs[‘‘sys_type_descr’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>Name:</font></td>
                    <td>
<?
  print $rs[‘‘sys_sysName’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System description:</font></td>                    <td>
<?
  print $rs[‘‘sys_sysDescr’’];
?>
                    </td>
                  </tr>
```

```
                    <tr>
                      <td><font class=‘‘text-bold’’>System location:</font></td>
                      <td>
<?
  print $rs[‘‘sys_sysLocation’’];
?>
                      </td>
                    </tr>
                    <tr>
                      <td><font class=‘‘text-bold’’>System uptime:</font></td>
                      <td>
<?
  print uptime($rs[‘‘sys_sysUpTime’’]);
?>
                      </td>
                    </tr>
                    <tr>
                      <td> </td>
                      <td> </td>
                    </tr>
                    <tr>
                      <td><font class=‘‘text-bold’’>System contact:</font></td>
                      <td>
<?
  print ‘‘<a href=\‘‘mailto:’’.$rs[‘‘sys_sysContact’’].’’\’’>’’.$rs[‘‘sys_sysContact’’].’’</a>’’;
?>
                      </td>
                    </tr>
                    <tr>
                      <td> </td>
                      <td> </td>
                    </tr>
                  </table>
```

```
              </td>
            </tr>
            <tr>
              <td>
<?
  network_information($sys_id);
?>
              </td>
            </tr>
          </table>
        </td>
        <td width=‘‘60%’’ align=‘‘left’’ valign=‘‘top’’>
          <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
            <tr>
              <td>
                <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                  <tr>
                    <td><hr></td>
                  </tr>
                  <tr>
                    <td align=‘‘center’’><font class=‘‘text-bold’’>Harddisk information</font></td>
                  </tr>
                  <tr>
                    <td><hr></td>
                  </tr>
                </table>
              </td>
            </tr>
          <tr>
            <td>
              <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                <tr>
                  <td nowrap><font class=‘‘text-bold’’>Mountpoint</font></td>
```

```
                  <td nowrap><font class=‘‘text-bold’’>Size (GB)</font></td>
                  <td nowrap><font class=‘‘text-bold’’>Left (GB)</font></td>
                  <td nowrap><font class=‘‘text-bold’’>Peak usage (GB)</font></td>
                  <td nowrap><font class=‘‘text-bold’’>Peak date</font></td>
                </tr>
<?
  while($rs2 = $hdds->getnext())
  {
    print ‘‘
      <tr>
        <td><font class=\‘‘text-normal\’’>’’.$rs2[‘‘comp_hdd_mpoint’’].’’</font></td>
        <td><font class=\‘‘text-normal\’’>’’.$rs2[‘‘comp_hdd_size’’].’’</font></td>
        <td ‘‘;
    $percent = $rs2[‘‘comp_hdd_left’’] / $rs2[‘‘comp_hdd_size’’] * 100;
    if($percent <= $_SESSION[‘‘hdd_left’’])
      print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
    print ‘‘
      ><font class=\‘‘text-normal\’’>’’.$rs2[‘‘comp_hdd_left’’].’’</font></td>
      <td><font class=\‘‘text-normal\’’>’’.$rs2[‘‘comp_hdd_peak’’].’’</font></td>
      <td><font class=\‘‘text-normal\’’>’’;
    $peak_date = date(‘‘Y-m-d H:i’’,$rs2[‘‘comp_hdd_peak_date’’]);
    print $peak_date;
    print ‘‘    </font>
        </td>
      </tr>’’;
  }
?>
              </table>
            </td>
          </tr>
          <tr>
            <td height=‘‘20’’> </td>
          </tr>
```

```
            <tr>
              <td>
                <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                  <tr>
                    <td><hr></td>
                  </tr>
                  <tr>
                    <td align=‘‘center’’><font class=‘‘text-bold’’>History information</font></td>
                  </tr>
                  <tr>
                    <td><hr></td>
                  </tr>
                </table>
              </td>
            </tr>
            <tr>
              <td>
                <table width=‘‘100%’’>
                  <tr>
                    <th>Date</th>
                    <th>Cpu-usage</th>
                    <th>Mem-usage</th>
                    <th>Temperature</th>
                  </tr>
<?
  while($rs3 = $srv_events->getnext())
  {
    print ‘‘
      <tr>
        <td>
          <font class=\‘‘font-normal\’’>’’;
    $date_event = date(‘‘Y-m-d H:i’’,$rs3[‘‘comp_e_date’’]);
    printv  $date_event.’’
```

```
      </font>

      </td>
      <td ‘‘;
      if($rs3[‘‘comp_e_cpuusage’’] >= $_SESSION[‘‘cpu_usage’’])
        print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
print     ’’>

      <font class=\‘‘font-normal\’’>
      ‘‘.$rs3[‘‘comp_e_cpuusage’’].’’  %
      </font>

      </td>
      <td ‘‘;
      if($rs3[‘‘comp_e_memusage’’] >= $_SESSION[‘‘mem_usage’’])
        print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
print    ’’>

      <font class=\‘‘font-normal\’’>
      ‘‘.$rs3[‘‘comp_e_memusage’’].’’  %
      </font>

      </td>
      <td ‘‘;
      if($rs3[‘‘comp_e_temp’’] >= $_SESSION[‘‘cpu_temp’’])
        print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
print    ’’>

      <font class=\‘‘font-normal\’’>
      ‘‘.$rs3[‘‘comp_e_temp’’].’’  C
      </font>

      </td>
```

```
        </tr>'';
  }
  ?>
        </table>

        </td>
      </tr>
    </table>

    </td>
  </tr>
</table>
</body>
</html>
```

/swh/class.php

```
<?
  include(‘‘../include/class.php’’);

  //Class for selecting main-information about switches and hubs
  class swhs extends superClass
  {
    function swhs ()
    {
      $this->superClass(‘‘SELECT * FROM syss INNER JOIN sys_types ON syss.sys_type_nr =
sys_types.sys_type_id WHERE sys_type_id = 4 OR sys_type_id = 5;’’);
    }
  }

  //Class for selecting specific information about a switch or hub
  class swh_specific extends superClass
  {
```

```
    function swh_specific ($sys_id)
    {
      $this->superClass(‘‘SELECT *, sys_types.sys_type_descr FROM syss INNER JOIN sys_types ON
syss.sys_type_nr = sys_types.sys_type_id WHERE sys_id = ‘‘.$sys_id.’’;’’);
    }
  }

  //Class for selecting the total number of ports on a switch or hub
  class swh_get_num_prts extends superClass
  {
    function swh_get_num_prts ($sys_id)
    {
      $this->superClass(‘‘SELECT COUNT(*) AS prts FROM swh_prts WHERE sys_nr = ‘‘.$sys_id.’’;’’);
    }
  }

  //Class for selecting the number of active ports on a switch or hub
  class swh_get_num_active extends superClass
  {
    function swh_get_num_active ($sys_id)
    {
      $this->superClass(‘‘SELECT COUNT(*) AS prts_active FROM swh_prts WHERE sys_nr = ‘‘.$sys_id.’’  AND
swh_prt_speed <> 0;’’);
    }
  }

  //Class for selecting all ports for a specific hub or switch
  class swh_prts extends superClass
  {
    function swh_prts ($sys_id)
    {
      $this->superClass(‘‘SELECT * FROM swh_prts WHERE sys_nr = ‘‘.$sys_id.’’;’’);
    }
```

```
  }
?>
```

/swh/main.php

```
<?
  session_start();
  include(‘‘class.php’’);
  $swhs = new swhs();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Switches & Hubs</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
    <script language=‘‘JavaScript’’>
      function open_swh_info()
      {
        window.open(‘‘../tom.php’’, ‘‘swh_info_window’’,
‘‘fullscreen=no,toolbar=no,status=yes,menubar=no,scrollbars=no,resizable=no,directories=no,location=no,width=780,height=560,left=
        document.swh_info_send.submit();
      }
    </script>
  </head>
  <body>
    <font class=‘‘text-bold’’>Switches & Hubs</font>
    <table width=‘‘100%’’>
      <tr>
        <th>Name</th>
        <th>Location</th>
        <th>Active ports</th>
      </tr>
```

```
<?
  while($rs = $swhs->getnext())
  {
    $num_prts = new swh_get_num_prts($rs[‘‘sys_id’’]);
    $rs_num_prts = $num_prts->getnext();

    $active_prts = new swh_get_num_active($rs[‘‘sys_id’’]);
    $rs_num_active = $active_prts->getnext();

    print ‘‘<tr>’’;

    print ‘‘<td>’’;
    print ‘‘<a href=\‘‘#\’’
onclick=\‘‘document.swh_info_send.sys_id.value=‘‘.$rs[‘‘sys_id’’].’’;open_swh_info();\’’>’’.$rs[‘‘sys_sysName’’].’’</a>’’;
    print ‘‘</td>’’;

    print ‘‘<td>’’;
    print $rs[‘‘sys_sysLocation’’];
    print ‘‘</td>’’;

    print ‘‘<td ‘‘;
    $percent = 100 - $rs_num_active[‘‘prts_active’’] / $rs_num_prts[‘‘prts’’] * 100;
    if($percent <= $_SESSION[‘‘port_usage’’])
      print ‘‘style=\‘‘background-color :  yellow; font-weight :  bold\’’’;
    print ’’>’’;
    print $rs_num_active[‘‘prts_active’’].’’  / ‘‘.$rs_num_prts[‘‘prts’’];
    print ‘‘</td>’’;

    print ‘‘</tr>’’;
  }
  ?>
    </table>
    <form name=‘‘swh_info_send’’ action=‘‘swh_info.php’’ target=‘‘swh_info_window’’ method=‘‘POST’’>
```

```
      <input type=‘‘hidden’’ name=‘‘sys_id’’ value=‘‘’’>
    </form>
  </body>
</html>
```

/swh/swh_info.php

```
<?
  session_start();
  include(‘‘class.php’’);

  $sys_id = addslashes($_POST[‘‘sys_id’’]);

  $information = new specific_information($sys_id);
  $rs = $information->getnext();

  $num_prts = new swh_get_num_prts($sys_id);
  $rs_num_prts = $num_prts->getnext();

  $active_prts = new swh_get_num_active($sys_id);
  $rs_num_active = $active_prts->getnext();

  $swh_prts = new swh_prts($sys_id);
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Specific information</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘1’’>
```

```
      <tr>
        <td width=‘‘50%’’ align=‘‘left’’ valign=‘‘top’’>
          <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td><font class=‘‘text-bold’’>System information:</font></td>
            </tr>
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td>
                <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                  <tr>
                    <td><font class=‘‘text-bold’’>Type of device:</font></td>
                    <td>
<?
  print $rs[‘‘sys_type_descr’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>Name:</font></td>
                    <td>
<?
  print $rs[‘‘sys_sysName’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System description:</font></td>
```

```
                    <td>
<?
  print $rs[‘‘sys_sysDescr’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System location:</font></td>
                    <td>
<?
  print $rs[‘‘sys_sysLocation’’];
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System uptime:</font></td>
                    <td>
<?
  print uptime($rs[‘‘sys_sysUpTime’’]);
?>
                    </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>Active ports:</font></td>
                    <td
<?
  $percent = 100 - $rs_num_active[‘‘prts_active’’] / $rs_num_prts[‘‘prts’’] * 100;
  if($percent <= 10)
    print ‘‘style=\‘‘background-color :  yellow; font-weight :  bold\’’’;
?>
                    >
<?
  print $rs_num_active[‘‘prts_active’’].’’  / ‘‘.$rs_num_prts[‘‘prts’’];
```

```
?>
                    </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td> </td>
                  </tr>
                  <tr>
                    <td><font class=‘‘text-bold’’>System contact:</font></td>
                    <td>
<?
  print ‘‘<a href=\‘‘mailto:’’.$rs[‘‘sys_sysContact’’].’’\’’>’’.$rs[‘‘sys_sysContact’’].’’</a>’’;
?>
                    </td>
                  </tr>
                  <tr>
                    <td> </td>
                    <td> </td>
                  </tr>
                </table>
              </td>
            </tr>
            <tr>
              <td>
<?
  network_information($sys_id);
?>
              </td>
            </tr>
          </table>
        </td>
        <td width=‘‘50%’’ align=‘‘left’’ valign=‘‘top’’>
          <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ style=‘‘width :  150 px’’>
```

```
            <tr>
              <td><hr></td>
            </tr>
            <tr>
              <td align=‘‘center’’><font class=‘‘text-bold’’>Port information</font></td>
            </tr>
            <tr>
              <td><hr></td>
            </tr>
          <table>
          <table cellspacing=‘‘0’’ cellpadding=‘‘0’’>
            <tr>
              <td align=‘‘center’’><font class=‘‘text-bold’’>Nr</font></td>
              <td><font class=‘‘text-bold’’>Speed</font></td>
              <td><font class=‘‘text-bold’’>Duplex</font></td>
            </tr>
<?
  while($rs2 = $swh_prts->getnext())
  {
    print ‘‘<tr>
        <td align=\‘‘center\’’ style=\‘‘width :  50 px\’’><font class=\‘‘text-normal\’’>
          ‘‘.$rs2[‘‘swh_prt_num’’].’’</font></td>
        <td style=\‘‘width :  50 px\’’><font class=\‘‘text-normal\’’>’’;
    if($rs2[‘‘swh_prt_speed’’] > 0)
      print $rs2[‘‘swh_prt_speed’’];
    else
    print ‘‘inactive’’;
    print ‘‘    </font></td>
        <td style=\‘‘width :  50 px\’’><font class=\‘‘text-normal\’’>’’;
    if($rs2[‘‘swh_prt_speed’’] != 0)
    {
      if($rs2[‘‘swh_prt_duplex’’] == 1)
        print ‘‘full’’;
```

```
      else
        print ‘‘half’’;
    }
    print ‘‘    </font></td>
        </tr>’’;
  }
?>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```

/wrk/class.php

```
<?
  include(‘‘../include/class.php’’);

  //Class for selecting main-information about workstations
  class wrks extends superClass
  {
    function wrks ()
    {
      $this->superClass(‘‘SELECT * FROM syss INNER JOIN sys_types ON syss.sys_type_nr =
sys_types.sys_type_id WHERE sys_type_id = 2;’’);
    }
  }

  //Class for selecting the last four events that have occured for all the workstations
  class wrks_events extends superClass
  {
    function wrks_events ()
```

```
    {
      $this->superClass(‘‘SELECT * FROM comp_es INNER JOIN syss ON syss.sys_id = comp_es.sys_nr INNER
JOIN sys_types ON sys_types.sys_type_id = syss.sys_type_nr WHERE sys_types.sys_type_id = 2 ORDER BY
comp_es.comp_e_date DESC LIMIT 4;’’);
    }
  }
?>
```

/wrk/events.php

```
<?
  session_start();   include(‘‘class.php’’);

  $wrks_events = new wrks_events();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Workstations - Events</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’>
      <tr>
        <th>Date</th>
        <th>Name</th>
        <th>Location</th>
        <th>CPU-usage</th>
        <th>Mem-usage</th>
        <th>Temperature</th>
      </tr>
<?
```

```
while($rs = $wrks_events->getnext())
{
  print ‘‘<tr>
      <td>
        <font class=\‘‘font-normal\’’>’’;
  $date_event = date(‘‘Y-m-d H:i’’,$rs[‘‘comp_e_date’’]);
  print $date_event.’’
        </font>
      </td>
      <td>
        <font class=\‘‘font-normal\’’>
        ‘‘.$rs[‘‘sys_sysName’’].’’
        </font>
      </td>
      <td>
        <font class=\‘‘font-normal\’’>
        ‘‘.$rs[‘‘sys_sysLocation’’].’’
        </font>
      </td>
      <td ‘‘;
  if($rs[‘‘comp_e_cpuusage’’] >= $_SESSION[‘‘cpu_usage’’])
    print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
  print  ’’>
        <font class=\‘‘font-normal\’’>
        ‘‘.$rs[‘‘comp_e_cpuusage’’].’’  %
        </font>
      </td>
      <td ‘‘;
  if($rs[‘‘comp_e_memusage’’] >= $_SESSION[‘‘mem_usage’’])
    print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
  print  ’’>
        <font class=\‘‘font-normal\’’>
        ‘‘.$rs[‘‘comp_e_memusage’’].’’  %
```

```
          </font>
        </td>
        <td ‘‘;
    if($rs[‘‘comp_e_temp’’] >= $_SESSION[‘‘cpu_temp’’])
      print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
    print  ’’>
          <font class=\‘‘font-normal\’’>
          ‘‘.$rs[‘‘comp_e_temp’’].’’  C
          </font>
        </td>
      </tr>’’;
  }
?>
    </table>
  </body>
</html>
```


/wrk/main.php

```
<?
  session_start();
  include(‘‘class.php’’);

  $wrks = new wrks();
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Workstations</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
    <script language=‘‘JavaScript’’>
      function open_wrk_info()
```

```
      {
        window.open(‘‘../tom.php’’, ‘‘wrk_info_window’’,
‘‘fullscreen=no,toolbar=no,status=yes,menubar=no,scrollbars=no,resizable=no,directories=no,location=no,width=780,height=560,left=
        document.wrk_info_send.submit();
      }
    </script>
  </head>
  <body>
    <font class=‘‘text-bold’’>Workstations</font>
    <table width=‘‘100%’’>
      <tr>
        <th>Name</th>
        <th>Location</th>
        <th>Current IP-Addresses</th>
      </tr>
<?
  while($rs = $wrks->getnext())
  {
    $nics = new nics($rs[‘‘sys_id’’]);

    print ‘‘<tr>’’;

    print ‘‘<td>’’;
    print ‘‘<a href=\‘‘#\’’
onclick=\‘‘document.wrk_info_send.sys_id.value=‘‘.$rs[‘‘sys_id’’].’’;open_wrk_info();\’’>’’.$rs[‘‘sys_sysName’’].’’</a>’’;
    print ‘‘</td>’’;

    print ‘‘<td>’’;
    print $rs[‘‘sys_sysLocation’’];
    print ‘‘</td>’’;

    print ‘‘<td>’’;
    while($rs2 = $nics->getnext())
```

```
    {
      $ips = new ips($rs2[‘‘sys_nic_id’’]);

      while($rs3 = $ips->getnext())
      {
        print $rs3[‘‘sys_ip_NetAddress’’].’’, ‘‘;
      }
    }
    print ‘‘</td>’’;

    print ‘‘</tr>’’;
  }
?>
    </table>
    <form name=‘‘wrk_info_send’’ action=‘‘wrk_info.php’’ target=‘‘wrk_info_window’’ method=‘‘POST’’>
      <input type=‘‘hidden’’ name=‘‘sys_id’’ value=‘‘’’>
    </form>
  </body>
</html>
```

/wrk/wrk_info.php

```
<?
  session_start();
  include(‘‘class.php’’);

  $sys_id = addslashes($_POST[‘‘sys_id’’]);

  $information = new specific_information($sys_id);
  $rs = $information->getnext();

  $hdds = new hdds($sys_id);
  $wrk_events = new comp_events($sys_id);
```

```
?>

<!DOCTYPE HTML PUBLIC ‘‘-//W3C//DTD HTML 4.01 Transitional//EN’’>
<html>
  <head>
    <title>Specific information</title>
    <link rel=‘‘stylesheet’’ href=‘‘../include/style_main.css’’ type=‘‘text/css’’>
  </head>
  <body>
    <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘1’’>
    <tr>
      <td width=‘‘40%’’ align=‘‘left’’ valign=‘‘top’’>
        <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
          <tr>
            <td><hr></td>
          </tr>
          <tr>
            <td><font class=‘‘text-bold’’>System information:</font></td>
          </tr>
          <tr>
            <td><hr></td>
          </tr>
          <tr>
            <td>
              <table cellspacing=‘‘0’’ cellpadding=‘‘0’’ border=‘‘0’’>
                <tr>
                  <td><font class=‘‘text-bold’’>Type of device:</font></td>
                  <td>
<?
  print $rs[‘‘sys_type_descr’’];
?>
                  </td>
                </tr>
```

```
                <tr>
                  <td><font class=‘‘text-bold’’>Name:</font></td>
                  <td>
<?
  print $rs[‘‘sys_sysName’’];
?>
                  </td>
                </tr>
                <tr>
                  <td><font class=‘‘text-bold’’>System description:</font></td>
                  <td>
<?
  print $rs[‘‘sys_sysDescr’’];
?>
                  </td>
                </tr>
                <tr>
                  <td><font class=‘‘text-bold’’>System location:</font></td>
                  <td>
<?
  print $rs[‘‘sys_sysLocation’’];
?>
                  </td>
                </tr>
                <tr>
                  <td><font class=‘‘text-bold’’>System uptime:</font></td>
                  <td>
<?
  print uptime($rs[‘‘sys_sysUpTime’’]);
?>
                  </td>
                </tr>
                <tr>
```

```
                  <td> </td>
                  <td> </td>
                </tr>
                <tr>
                  <td><font class=‘‘text-bold’’>System contact:</font></td>
                  <td>
<?
  print ‘‘<a href=\‘‘mailto:’’.$rs[‘‘sys_sysContact’’].’’\’’>’’.$rs[‘‘sys_sysContact’’].’’</a>’’;
?>
                  </td>
                </tr>
                <tr>
                  <td> </td>
                  <td> </td>
                </tr>
              </table>
            </td>
          </tr>
          <tr>
            <td>
<?
  network_information($sys_id);
?>
            </td>
          </tr>
        </table>
      </td>
      <td width=‘‘60%’’ align=‘‘left’’ valign=‘‘top’’>
        <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
          <tr>
            <td>
              <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                <tr>
```

```
                    <td><hr></td>
                  </tr>
                  <tr>
                    <td align=‘‘center’’><font class=‘‘text-bold’’>Harddisk information</font></td>
                  </tr>
                  <tr>
                    <td><hr></td>
                  </tr>
                </table>
              </td>
            </tr>
            <tr>
              <td>
                <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                  <tr>
                    <td nowrap><font class=‘‘text-bold’’>Mountpoint</font></td>
                    <td nowrap><font class=‘‘text-bold’’>Size (GB)</font></td>
                    <td nowrap><font class=‘‘text-bold’’>Left (GB)</font></td>
                    <td nowrap><font class=‘‘text-bold’’>Peak usage (GB)</font></td>
                    <td nowrap><font class=‘‘text-bold’’>Peak date</font></td>
                  </tr>
<?
  while($rs2 = $hdds->getnext())
  {
    print ‘‘<tr>
        <td>
          <font class=\‘‘text-normal\’’>’’.
            $rs2[‘‘comp_hdd_mpoint’’]
          .’’</font>
        </td>
        <td>
          <font class=\‘‘text-normal\’’>
            ‘‘.$rs2[‘‘comp_hdd_size’’].’’
```

```
            </font>
          </td>
          <td ‘‘;
            $percent = $rs2[‘‘comp_hdd_left’’] / $rs2[‘‘comp_hdd_size’’] * 100;
            if($percent <= $_SESSION[‘‘hdd_left’’])
              print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
    print  ’’>
            <font class=\‘‘text-normal\’’>
              ‘‘.$rs2[‘‘comp_hdd_left’’].’’
            </font>
          </td>
          <td>
            <font class=\‘‘text-normal\’’>
              ‘‘.$rs2[‘‘comp_hdd_peak’’].’’
            </font>
          </td>
          <td>
            <font class=\‘‘text-normal\’’>’’;
              $peak_date = date(‘‘Y-m-d H:i’’,$rs2[‘‘comp_hdd_peak_date’’]);
    print $peak_date;
    print ‘‘    </font>
          </td>
          </tr>’’;
      }
    ?>
                </table>
              </td>
            </tr>
            <tr>
              <td height=‘‘20’’> </td>
            </tr>
            <tr>
              <td>
```

```
                <table width=‘‘100%’’ cellspacing=‘‘0’’ cellpadding=‘‘0’’>
                  <tr>
                    <td><hr></td>
                  </tr>
                  <tr>
                    <td align=‘‘center’’><font class=‘‘text-bold’’>History information</font></td>
                  </tr>
                  <tr>
                    <td><hr></td>
                  </tr>
                </table>
              </td>
            </tr>
            <tr>
              <td>
                <table width=‘‘100%’’>
                  <tr>
                    <th>Date</th>
                    <th>CPU-usage</th>
                    <th>Mem-usage</th>
                    <th>Temperature</th>
                  </tr>
<?
  while($rs3 = $wrk_events->getnext())
  {
    print ‘‘<tr>
        <td>
          <font class=\‘‘font-normal\’’>’’;
            $date_event = date(‘‘Y-m-d H:i’’,$rs3[‘‘comp_e_date’’]);
    print       $date_event.’’
          </font>
          </td>
```

```
            <td ‘‘;
            if($rs3[‘‘comp_e_cpuusage’’] >= $_SESSION[‘‘cpu_usage’’])
              print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
      print     ’’>

            <font class=\‘‘font-normal\’’>
            ‘‘.$rs3[‘‘comp_e_cpuusage’’].’’  %
            </font>

            </td>
            <td ‘‘;
            if($rs3[‘‘comp_e_memusage’’] >= $_SESSION[‘‘mem_usage’’])
              print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
      print    ’’>

            <font class=\‘‘font-normal\’’>
            ‘‘.$rs3[‘‘comp_e_memusage’’].’’  %
            </font>

            </td>
            <td ‘‘;
            if($rs3[‘‘comp_e_temp’’] >= $_SESSION[‘‘cpu_temp’’])
              print ‘‘style=\‘‘background-color :  red; font-weight :  bold\’’’;
      print    ’’>

            <font class=\‘‘font-normal\’’>
            ‘‘.$rs3[‘‘comp_e_temp’’].’’  C
            </font>

            </td>
          </tr>’’;
  }
?>
```

```
                </table>
              </td>
            </tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```